



TUGAS AKHIR - SM141501

**MEDIA BANTU BERBASIS ANDROID
UNTUK MEMBERIKAN INFORMASI
KANDUNGAN HARAM DALAM PRODUK
MAKANAN DAN MINUMAN DENGAN
MENGUNAKAN METODE *DECISION
TREE***

**HAIDAR AMMARAFIF RAMZY
NRP 1213 100 117**

**Dosen Pembimbing
Prof. Dr. Mohammad Isa Irawan, MT**

**DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

“Halaman ini sengaja dikosongkan”



FINAL PROJECT - SM141501

**ANDROID-BASED MEDIA TOOLS TO
PROVIDE INFORMATION HARAM
CONTENT IN FOOD PRODUCT AND
BEVERAGES USING DECISION TREE
METHOD**

**HAIDAR AMMARAFIF RAMZY
NRP 1213 100 117**

**Supervisor
Prof. Dr. Mohammad Isa Irawan, MT**

**DEPARTMENT OF MATHEMATICS
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

“Halaman ini sengaja dikosongkan”

LEMBAR PENGESAHAN

**MEDIA BANTU BERBASIS ANDROID UNTUK
MEMBERIKAN INFORMASI KANDUNGAN HARAM
DALAM PRODUK MAKANAN DAN MINUMAN DENGAN
MENGUNAKAN METODE *DECISION TREE***

***ANDROID-BASED MEDIA TOOLS TO PROVIDE
INFORMATION HARAM CONTENT IN FOOD PRODUCT
AND BEVERAGES USING DECISION TREE METHOD***

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang minat Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

Oleh:

HAIDAR AMMARAFIF RAMZY
NRP. 1213 100 117

Menyetujui,
Dosen Pembimbing,

Prof. Dr. Mohammad Isa Irawan, MT
NIP. 19631225 198903 1 001

Mengetahui,
Kepala Departemen Matematika
FMIPA ITS

Dr. Imam Mukhlash, S.Si, M.T
NIP. 19700831 199403 1 003

“Halaman ini sengaja dikosongkan”

MEDIA BANTU BERBASIS ANDROID UNTUK MEMBERIKAN INFORMASI KANDUNGAN HARAM DALAM PRODUK MAKANAN DAN MINUMAN DENGAN MENGUNAKAN METODE *DECISION TREE*

Nama Mahasiswa : Haidar Ammarafif Ramzy
NRP : 1213 100 117
Departemen : Matematika
Dosen Pembimbing : Prof. Dr. M. Isa Irawan, MT

ABSTRAK

Semakin pesatnya perkembangan produk makanan dan minuman di Indonesia, membuat produsen berlomba-lomba untuk membuat produk makanan dan minuman yang digemari oleh masyarakat. Berbagai jenis makanan dan minuman yang dihasilkan, dibuat dengan bermacam-macam *ingredients*. Hal ini membuat konsumen muslim semakin sulit dan berhati-hati untuk menentukan produk makanan dan minuman yang halal sesuai syariat Islam. Oleh karena itu, perlu adanya alat bantu untuk mengetahui dan memberikan informasi secara cepat kandungan haram dari suatu produk makanan dan minuman.

Semakin populernya pemanfaatan gadget bersistem operasi Android merupakan peluang untuk bisa dimanfaatkan untuk tujuan tersebut. Metode yang akan diimplementasikan dalam program alat bantu berbasis Android tersebut adalah *Decision Tree*. Metode *Decision Tree* merupakan metode klasifikasi yang paling populer digunakan. Selain karena pembangunannya relatif cepat, hasil dari model yang dibangun mudah untuk dipahami.

Dari pengujian oleh beberapa mahasiswa. Hasil penelitian ini dapat membantu dalam memilih produk makanan dan minuman yang halal, serta dapat membantu menghindari produk makanan dan minuman yang haram.

***Kata kunci : Halal,Haram,Ingredients,E-Number,Decision Tree,
Klasifikasi***

“Halaman ini sengaja dikosongkan”

**ANDROID-BASED MEDIA TOOLS TO PROVIDE
INFORMATION HARAM CONTENT IN FOOD PRODUCTS
AND BEVERAGES USING DECISION TREE METHOD**

Name of Student : Haidar Ammarafif Ramzy
NRP : 1213 100 117
Department : Mathematics
Supervisor : Prof. Dr. M. Isa Irawan, MT

ABSTRACT

The more rapid development of food and beverage products in Indonesia, make producers competing for food and beverage products are favored by the community. Various types of food and beverages produced, made with a variety of materials. This is in accordance with Islamic Shari'a. Therefore, the need for tools to know and provide information quickly the illicit content of a food and beverage products.

The increasing popularity of gadget utilized Android operating system is an opportunity to be utilized for the purpose. The method to be implemented in the Android based tool program is Decision Tree. The Decision Tree method is the most popular list method used. In addition to its relatively rapid development, the results of the built model are easy to understand.

From testing by some students. The results of this study can help in selecting halal food and beverage products, and can help avoid unlawful food and beverage products.

**Keywords : Keywords : Halal,Haram,Ingredients,E-
Number,Decision Tree,Classification**

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Segala puji syukur penulis panjatkan ke hadirat Allah SWT, karena dengan ridlo-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul

“MEDIA BANTU BERBASIS ANDROID UNTUK MEMBERIKAN INFORMASI KANDUNGAN HARAM DALAM PRODUK MAKANAN DAN MINUMAN MENGUNAKAN METODE *DECISION TREE*”

yang merupakan salah satu persyaratan akademis dalam menyelesaikan Program Sarjana Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat diselesaikan dengan baik berkat kerja sama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal tersebut, penulis ingin mengucapkan terima kasih kepada :

1. Dr. Imam Mukhlash, S.Si, MT selaku Kepala Departemen Matematika ITS dan selaku Dosen Wali yang telah memberikan arahan akademik selama penulis menempuh Pendidikan di departemen Matematika ITS.
2. Prof. Dr. Mohammad Isa Irawan, MT selaku Dosen Pembimbing yang telah memberikan bimbingan dan motivasi kepada penulis dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.
3. Dr. Didik Khusnul Arif, S.Si, M.Si selaku Kepala Program Studi S1 Departemen Matematika ITS.
4. Drs. Iis Herisman, M.Si selaku Sekretaris Program Studi S1 Departemen Matematika ITS.
5. Seluruh jajaran dosen dan staf Departemen Matematika ITS.
6. Keluarga tercinta yang senantiasa memberikan dukungan dan do'a yang tak terhingga.
7. Teman-teman angkatan 2013 yang saling mendukung dan memotivasi.
8. Semua pihak yang tak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Agustus 2017

Penulis

DAFTAR ISI

	Hal
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah	5
1.4 Tujuan	5
1.5 Manfaat	6
1.6 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA	
2.1 Penelitian Terdahulu	9
2.2 Perbandingan Aplikasi	10
2.3 Food Ingredients Numbers (E-Numbers).....	12
2.4 Android Studio.....	13
2.5 Android	14
2.5 SQLite	17
2.6 Decision Tree	19
2.7 Rule-Based Decision Tree.....	20
BAB III METODOLOGI PENELITIAN	
3.1 Tahapan Penelitian.....	21
3.2 Diagram Alir Penelitian	23
BAB IV PERANCANGAN APLIKASI DAN IMPLEMENTASI	
4.1 Perancangan Aplikasi.....	25
4.2 Analisis Kebutuhan Sistem	27
4.3 Analisis Kebutuhan User	28
4.4 Desain Sistem.....	28

4.5	Desain Proses Decision Tree	31
4.6	Analisis Aplikasi	35
4.7	Perancangan User Interface.....	43
BAB V UJI COBA DAN PEMBAHASAN		
5.1	Pengujian Halaman Utama.....	45
5.2	Pengujian Menu Cek Produk	46
5.3	Pengujian Daftar Ingredients.....	47
5.4	Pengujian Menu Lihat Hasil Cek Produk.....	48
5.5	Pengujian Proses Decision Tree	49
BAB VI PENUTUP		
6.1	Kesimpulan	51
6.2	Saran.....	51
DAFTAR PUSTAKA.....		52
LAMPIRAN		55
BIODATA PENULIS		80

DAFTAR GAMBAR

	Hal
Gambar 2.1 Android LifeCycle	17
Gambar 2.2 Diagram Rule-Based Decision Tree	20
Gambar 3.1 Diagram Alir Penelitian	24
Gambar 4.1 CDM pada Aplikasi Android	27
Gambar 4.2 Desain Sistem Perangkat Lunak	30
Gambar 4.3 Desain Decision Tree	31
Gambar 4.4 Desain Tree saat produk halal	33
Gambar 4.5 Desain Tree saat produk hram	34
Gambar 4.6 Use Case Diagram User	37
Gambar 4.7 Activity Diagram	38
Gambar 4.8 Data Flow Diagram Level 0	39
Gambar 4.9 Data Flow Diagram Level 1	40
Gambar 4.10 Data Flow Diagram Level 2	41
Gambar 4.11 DFD Level 2 untuk proses decision tree	41
Gambar 4.12 DFD Level 2 untuk menampilkan hasil produk halal/haram	42
Gambar 5.1 Tampilan Awal	45
Gambar 5.3 TextField AutoComplete	47
Gambar 5.2 Menu Cek Produk	47
Gambar 5.5 Menu daftar ingredients setelah di klik	48
Gambar 5.4 Menu daftar ingredients	48
Gambar 5.6 Tampilan List Hasil	49
Gambar 5.7 Tampilan Hasil Setelah di klik	49
Gambar 5.10 Proses Decision Tree	50
Gambar 5.9 Pengujian aplikasi	50

“Halaman ini sengaja dikosongkan”

DAFTAR TABEL

Tabel 1.1 Populasi agama di Indonesia [3].....	1
Tabel 2.1. Konvensi untuk menetapkan kode E - Number..	13

“Halaman ini sengaja dikosongkan”

BAB I

PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang yang mendasari penulisan Tugas Akhir ini. Di dalamnya mencakup identifikasi permasalahan pada topik Tugas Akhir kemudian dirumuskan menjadi permasalahan yang diberikan batasan-batasan dalam pembahasan pada Tugas Akhir ini.

1.1 Latar Belakang

Indonesia merupakan negara dengan mayoritas penduduk beragama Islam terbesar di Asia Tenggara yakni sebesar 207.176.162 [1]. Populasi muslim di Indonesia menduduki peringkat tertinggi, hal ini dapat ditunjukkan pada Tabel 1. Indonesia dengan populasi penduduk mayoritas Muslim merupakan pasar potensial yang besar bagi berbagai produsen barang dan jasa. Konsumen muslim di Indonesia mencari sertifikasi halal otentik yang dikeluarkan oleh MUI (Majelis Ulama Indonesia). Sertifikasi ini memberikan kewenangan bagi perusahaan dalam penggunaan logo halal untuk dicetak pada kemasan produk atau untuk dipajang di premis perusahaan [2].

Tabel 1.1 Populasi agama di Indonesia [3]

Agama	Angka Absolut (dalam juta)	Persentase (dari populasi total)
Islam	207,2	87,2
Kristen	16,5	6,9
Katolik	6,9	2,9
Hindu	4,0	1,7
Buddha	1,7	0,7
Konghucu	0,1	0,05
TOTAL	236.4	100

Dilihat dari perspektif Islam, konsep halal merupakan hal yang vital bagi seorang muslim. Halal berarti diperbolehkan atau diijinkan dalam agama Islam, dalam surah Al-Baqarah ayat 168 yang berbunyi:

يَتَأْتِيهَا النَّاسُ كُلُّوْا مِمَّا فِي الْأَرْضِ حَلٰلًا طَيِّبًا وَلَا تَتَّبِعُوْا خُطُوٰتِ
الشَّيْطٰنِ ۚ إِنَّهُ لَكُمْ عَدُوٌّ مُّبِيْنٌ ﴿١٦٨﴾

Artinya:

“Hai sekalian manusia, makanlah yang halal lagi baik dari apa yang terdapat di bumi, dan janganlah kamu mengikuti langkah-langkah syaitan; karena sesungguhnya syaitan itu adalah musuh yang nyata bagimu”

Dan dijelaskan juga jenis-jenis makanan yang diharamkan dalam islam pada surah Al-Baqarah ayat 173 yang berbunyi:

إِنَّمَا حَرَّمَ عَلَيْكُمُ الْمَيْتَةَ وَالدَّمَ وَلَحْمَ الْخِنْزِيْرِ وَمَا أَهْلَ بِهِ
لِغَيْرِ اللَّهِ ۖ فَمَنْ أَضْطَرَّ غَيْرَ بَاغٍ وَلَا عَادٍ فَلَا إِثْمَ عَلَيْهِ ۚ إِنَّ اللَّهَ غَفُوْرٌ
رَّحِيْمٌ ﴿١٧٣﴾

Artinya:

“Sesungguhnya Allah hanya mengharamkan bagimu bangkai, darah, daging babi, dan binatang yang (ketika disembelih) disebut (nama) selain Allah. Tetapi barangsiapa dalam keadaan terpaksa (memakannya) sedang dia tidak menginginkannya dan tidak (pula) melampaui batas, maka tidak ada dosa baginya. Sesungguhnya Allah Maha Pengampun lagi Maha Penyayang”

Berdasarkan dalil-dalil Al-Quran tersebut, muslim akan mencari produk untuk dikonsumsi sesuai dengan ajaran agama yang tertulis dalam Al-Quran. Hal ini ditandai dengan banyaknya permintaan produk halal yang sudah memiliki sertifikat Halal di dunia [4].

Seiring dengan meningkatnya permintaan produk halal, studi dalam konteks pengakuan terhadap makanan halal perlu diperiksa lebih lanjut. Produk halal harus diakui sebagai simbol kebersihan, keamanan, dan kualitas tinggi bagi konsumen muslim. Bahan makanan (*food ingredients*) adalah bahan-bahan yang digunakan dalam membuat suatu produk makanan. Bahan makanan menjadi salah satu faktor penentu dalam memprediksi bagaimana minat beli konsumen terhadap suatu produk makanan. Menurut penelitian Yunus, Rashid, dan Ariffin (2014) terkait minat membeli produk halal menunjukkan bahwa komposisi bahan-bahan yang terdapat dalam produk berpengaruh positif secara signifikan terhadap minat membeli konsumen muslim[5].

Seiring dengan berkembangnya teknologi informasi menyebabkan semakin banyak orang mengerti akan pentingnya fungsi komputer dalam membantu pekerjaan mereka. Saat ini perkembangan komputer telah membawa perubahan besar dalam berbagai bidang diantaranya bidang Politik, Ilmu Pengetahuan, Ekonomi, Sosial, Budaya, dan Kesehatan. Perkembangan teknologi informasi, selain perkembangan aplikasi desktop pada komputer juga meliputi perkembangan aplikasi mobile. Semakin berkembangnya aplikasi mobile maka, terciptalah sebuah sistem operasi yang dikembangkan untuk perangkat mobile berbasis linux yaitu Android.

Pada saat ini banyak masyarakat yang menggunakan *mobile phone* untuk kemudahan dalam berkomunikasi dan akses internet. Dalam pengembangan sebuah sistem operasi *mobile-phone* saat ini yang sering kita jumpai yaitu sistem operasi android. *Android* dinilai lebih efektif, Karena kecepatan dan kemudahannya dalam mengakses sebuah informasi. Maka dari permasalahan dan fakta tersebut, perlu dicarikan suatu upaya untuk pemecahannya. Salah satunya adalah dengan membangun sistem informasi untuk membantu pengguna/*user* dalam menentukan kehalalan suatu produk.

Decision tree merupakan salah satu metode *Classification* yang menggunakan representasi struktur pohon (tree) di mana setiap node merepresentasikan atribut, cabangnya merepresentasikan nilai dari atribut, dan daun merepresentasikan kelas. Node yang paling atas dari decision tree disebut sebagai root. *Decision tree* merupakan metode *Classifiaction* yang paling populer digunakan. Selain karena pembangunannya relatif cepat, hasil dari model yang dibangun mudah untuk dipahami[6].

Mencermati hal-hal di atas maka dalam proposal penelitian ini akan diusulkan sebuah alat bantu berbasis android untuk mendeteksi kandungan haram dalam produk makanan dan minuman dengan metode *decision tree*. Untuk memudahkan pengguna untuk menentukan kehalalan suatu produk menggunakan *mobile phone*, maka pengguna tidak perlu lagi ragu untuk membeli produk-produk makanan dan minuman yang dijual di *supermarket* atau toko-toko penjual makanan lainnya. Pada penelitian ini faktor-faktor yang dibutuhkan dalam menentukan kehalalan suatu produk adalah merek makanan dan minuman, *ingredients* suatu produk, dan kategori kehalalan.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, dapat dirumuskan permasalahan dalam Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana merancang media bantu berbasis android untuk memberikan informasi kandungan haram dalam produk makanan dan minuman dengan metode *decision tree*?
2. Bagaimana implementasi metode *decision tree* untuk menentukan kandungan haram?

1.3 Batasan Masalah

Pada penelitian ini, penulis membuat batasan masalah sebagai berikut :

1. Dataset yang digunakan diperoleh dari *European Economic Community (ECC)*.
2. Menggunakan *software* Android Studio.
3. Menggunakan DBMS (*Data Base Management System*) yaitu SQLite untuk pengolahan data.
4. Pengujian hanya pada makanan buatan pabrik.
5. *Input* hanya *ingredients* yang tak dikenal.
6. Data *ingredients* haram yang berasal dari babi, alkohol, dan bukan dari bangkai dan binatang yang disembelih bukan atas nama Allah.

1.4 Tujuan

Berdasarkan permasalahan yang telah dirumuskan sebelumnya, tujuan penelitian Tugas Akhir ini adalah :

1. Merancang media bantu berbasis android untuk memberikan informasi kandungan haram dalam produk makanan dan minuman dengan metode *decision tree*.
2. Mendapatkan hasil implementasi dari metode *decision tree* dalam menentukan kandungan haram dari produk makanan dan minuman.

1.5 Manfaat

Manfaat dari Tugas Akhir ini adalah:

1. Dengan adanya aplikasi ini, dapat meningkatkanantisipasi terhadap produk makanan dan minuman yang belum bersertifikasi halal.
2. Memberikan informasi kepada pengguna tentang kehalalan suatu produk serta membantu pengambilan keputusan dalam menentukan produk makanan dan minuman yang halal.

1.6 Sistematika Penulisan

Sistematika dari penulisan Tugas Akhir ini adalah sebagai berikut :

1. BAB I PENDAHULUAN

Bab ini berisi tentang gambaran umum dari penulisan Tugas Akhir yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang teori dasar yang mendukung dalam Tugas Akhir ini, yaitu penelitian terdahulu, meliputi penelitian sebelumnya terkait Tugas Akhir ini, terdiri dari penelitian terdahulu, struktur *framework Android Studio*, DBMS (*Data Base Management System*), SQLite, *Decision Tree* dan *Rule-Based Classifiers*.

3. BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tahapan-tahapan dan metode yang digunakan untuk menyelesaikan Tugas Akhir ini.

4. BAB IV PERANCANGAN APLIKASI DAN IMPLEMENTASI

Pada bab ini akan menguraikan bagaimana tahapan tahapan dalam perancangan implementasi. Pembahasan perancangan implementasi dimulai dari perancangan pra-pemrosesan data.

5. BAB V UJI COBA DAN PEMBAHASAN

Bab ini menjelaskan mengenai uji coba aplikasi yang sudah dibuat melalui perangkat Android. Implementasi pada setiap proses, dan melakukan pembahasan terhadap hasil yang diperoleh.

6. BAB VI PENUTUP

Bab ini berisi kesimpulan yang diperoleh dari pembahasan masalah sebelumnya serta saran yang diberikan untuk pengembangan selanjutnya.

“Halaman ini sengaja dikosongkan”

BAB II TINJAUAN PUSTAKA

Pada bagian ini dijelaskan menjelaskan tentang kajian teori dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam tugas akhir ini, meliputi penelitian sebelumnya terkait Tugas Akhir ini, terdiri dari penelitian terdahulu, struktur *framework Android Studio*, DBMS (*Data Base Management System*), *SQLite*, *Decision Tree* dan *Rule-Based Classifiers*.

2.1 Penelitian Terdahulu

Survey terhadap aplikasi sejenis adalah aplikasi Halal MUI. Aplikasi Halal MUI menggunakan *scan barcode* untuk pengecekannya. Dan aplikasi ini menggunakan database produk-produk makanan dan minuman yang didapat dari LPPOM MUI. Produk makanan yang masuk ke Indonesia akan diperiksa melalui LPPOM MUI lalu diberi keterangan halal atau haram. Data itulah yang digunakan aplikasi ini untuk *scan barcode* pada suatu produk makanan dan minuman.

Penelitian terkait tugas akhir ini yang sudah dilakukan oleh Muhammad Athoillah pada tahun 2013 yang berjudul Perancangan Sistem Informasi *Mobile* Berbasis Android Untuk Kontrol Persediaan Barang di Gudang. Penelitian ini membuat aplikasi berbasis android dengan menggunakan sistem informasi yang terhubung langsung dengan *database computer server*, sehingga memudahkan penggunaanya untuk mengakses database yang tersedia secara mobile[7].

Penelitian selanjutnya terkait tugas akhir ini yang sudah dilakukan oleh M Ridho Bintang J pada tahun 2015 yang berjudul Perancangan Prototipe Sistem Pakar Berbasis Android untuk Pendukung Keputusan Diagnosa Kerusakan pada Mobil di Auto200. Penelitian ini membuat aplikasi berbasis android dengan sistem pakar menggunakan metode pencarian *Forward*


Chaining, yaitu metode yang menghasilkan kesimpulan dari seperangkat data yang diketahui seperti jenis kendaraan, keluhan, kilometer dan faktor kerusakan [8].

2.2 Perbandingan Aplikasi

Aplikasi-aplikasi mengenai pengecekan produk halal atau haram sudah ada sebelumnya. Namun beberapa dari aplikasi tersebut masih belum bisa mengetahui produk yang diperiksa halal atau haram karena menggunakan scan barcode yang tidak dikenali. Berikut adalah aplikasi-aplikasi sejenis yang sudah ada :

No	Nama	Logo	Keterangan
1	HALAL MUI		<ul style="list-style-type: none"> - Pengecekan utama menggunakan <i>scan barcode</i> - Pengecekan menggunakan nama produk, produsen dan sertifikat. - Menggunakan database produk dari LPPOM MUI - Tidak ada pengecekan melalui <i>ingredients</i>
2	Scan Halal		<ul style="list-style-type: none"> - Pengecekan utama menggunakan <i>scan barcode</i> - Terdapat menu <i>trending product</i> yang berfungsi untuk melihat produk yang sering di periksa

			<ul style="list-style-type: none"> - Produk yang diperiksa mayoritas berasal dari luar negeri - Tidak ada pengecekan melalui <i>ingredients</i>
3	Halal Trip		<ul style="list-style-type: none"> - Aplikasi hanya berupa <i>social media</i> untuk share makanan halal di seluruh dunia - Tidak ada pengecekan produk
4	Halal E-Numbers		<ul style="list-style-type: none"> - Aplikasi berisi data <i>ingredients e-numbers</i> - Pengecekan menggunakan input kode <i>ingrdients</i> - Tidak ada pengecekan produk
5	Halal Indonesia		<ul style="list-style-type: none"> - Aplikasi ini menggunakan database dari LPPOM MUI - Pengecekan pada makan halal, restoran dan kosmetik. - Terdapat pencarian menggunakan <i>E-Numbers</i>
6	CekHalal		<ul style="list-style-type: none"> - Pencarian dengan menginputkan nama produk - Database di-<i>update</i> hanya sampai Mei 2015
7	Halal Check E-Numbers		<ul style="list-style-type: none"> - Pengecekan menggunakan input kode E-Numbers

			- Tidak ada pengecekan produk
8	HalalMinds Find Halal Product		<ul style="list-style-type: none"> - Pengecekan menggunakan <i>scan barcode</i> - Terdapat menu untuk pencarian restoran yang halal - Terdapat pencarian menggunakan <i>E-Number</i>

2.3 *Food Ingredients Numbers (E-Numbers)*

E-Numbers merepresentasi bahan tambahan makanan tertentu, yang digunakan oleh industri makanan dalam pembuatan berbagai produk makanan. *E-Numbers* ini telah dirumuskan oleh *European Economic Community* (EEC) dan diadopsi secara universal oleh industri makanan di seluruh dunia. *E-Numbers* ini adalah referensi nomor yang digunakan oleh Uni Eropa untuk memfasilitasi identifikasi komposisi makanan. Semua bahan tambahan makanan yang digunakan di Uni Eropa diidentifikasi oleh *E-Numbers*. "E" singkatan dari "Eropa" atau "Uni Eropa". Biasanya setiap bahan tambahan makanan diberikan sebuah nomor yang unik, meskipun kadang-kadang, bahan tambahan yang terkait diberikan perpanjangan ("a", "b", atau "i", "ii") ke dalam *E-Numbers* yang lain.

Komisi Uni Eropa menetapkan *E-Numbers* setelah bahan tambahan dihapus oleh *Scientific Committee on Food* (SCF), badan yang bertanggung jawab untuk evaluasi keamanan bahan tambahan makanan di Uni Eropa. Konvensi untuk menetapkan *E-Number* bisa dilihat pada Tabel 2.1. *E-Number* tersebut hanya digunakan untuk bahan yang ditambahkan secara langsung ke produk makanan, sehingga

kontaminan, enzim dan alat bantu pengolahan, yang dapat diklasifikasikan sebagai bahan tambahan, tidak termasuk dalam sistem *E-Number* [9]. Untuk *data set ingredients* halal haram dapat dilihat pada Lampiran 1.

Tabel 2.1. Konvensi untuk menetapkan kode E - *Number*

Nomor	Keterangan
100-199	Pewarna Makanan
200-299	Pengawet
300-399	Antioksidan, fosfat, dan bahan pengompleks
400-499	Pengental, bahan pembentuk gel, fosfat, humektan, bahan pengemulsi
500-599	Garam dan senyawa terkait
600-699	Penyedap rasa
700-899	Tidak digunakan untuk bahan tambahan makanan
900-999	Bahan pelapis permukaan, gas, pemanis
1000-1399	Bahan tambahan lainnya
1400-1499	Turunan zat tepung

2.4 Android Studio

Android Studio adalah sebuah IDE untuk *Android Development* yang diperkenalkan google pada acara *Google I/O 2013*. *Android Studio* merupakan pengembangan dari *Eclipse IDE*, dan dibuat berdasarkan *IDE Java* populer, yaitu *IntelliJ IDEA*. *Android Studio* merupakan *IDE* resmi untuk pengembangan aplikasi *Android* [10].

2.5 Android

Android adalah sistem operasi telepon selular berbasis kernel linux, android merupakan sistem operasi berplatform terbuka sehingga setiap orang bias mengembangkan sistem operasi ini secara bebas, android merupakan *mobile device* yang dianggap memiliki kemampuan yang lebih peka terhadap lokasi dan preferensi pemiliknya.

Awalnya, perusahaan *search engine* terbesar saat ini, yaitu google inc. Membeli Android Inc. Pendatang baru yang membuat perangkat untuk ponsel. Kemudian untuk mengembangkan Android dibentuklah Open Handset Alliance konsorsium dari 34 perusahaan hardware, software dan telekomunikasi, termasuk diantaranya yaitu Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Android menjadi pesaing utama produk *Smartphone*. Pesatnya pertumbuhan android Karena android adalah platform yang sangat lengkap baik dari segi sistem operasi, aplikasi dan tool pengembangannya, sehingga android berkembang pesat hingga saat ini, baik dari segi teknologi maupun dari segi jumlah *device* di dunia.

Sampai saat ini android telah banyak berkembang hingga beberapa versi. Android versi 1.1 adalah versi pertama yang dirilis pada tahun 2009, kemudian berurut – turut muncul versi yang lain yang merupakan perbaikan demi perbaikan dari versi yang sebelumnya diantaranya ialah versi 1.5 (Cupcake), versi 1.6 (Donut), versi 2.0/2.1 (Éclair), versi 2.2 (Froyo), versi 2.3 (Gingerbread), versi 3.0/3.1 (Honeycomb), versi 4.0 (ICS), versi 4.1/4.3 (Jelly Bean), versi 4.4 (KitKat), versi 5.0 (Lollipop), versi 6.0 (Marshmallow), hingga yang terbaru saat ini adalah versi 7.0 (Nougat) [10].

2.4.1 Android Activity

Di Android, *Activity* ini merupakan komponen yang menyediakan halaman tampilan agar *user* dapat berinteraksi dengan fitur dan aplikasi. Karena biasanya setiap aplikasi terdiri dari beberapa *activity*, dan tentunya memiliki *activity* utama. Dimana *activity* utama dapat dilihat saat kita pertama kali membuka aplikasi. *Activity* utama inilah yang nantinya mengarahkan user ke *activity-activity* lainnya[10]. Berikut adalah *activity* yang ada dalam android:

- 1) *Running*: *Activity* ini terlihat, dan *user* sedang berinteraksi dengan *Activity* ini.
- 2) *Paused*: *Activity* ini masih terlihat, tetapi tidak dalam proses bekerja.
- 3) *Stopped*: *Activity* ini tidak terlihat, dan kemungkinan akan di *kill* oleh sistem.

Killed: *Activity* ini telah diakhiri/ditutup oleh sistem dengan memanggil *method* **its**`finish()`.

2.4.2 Method pada Android

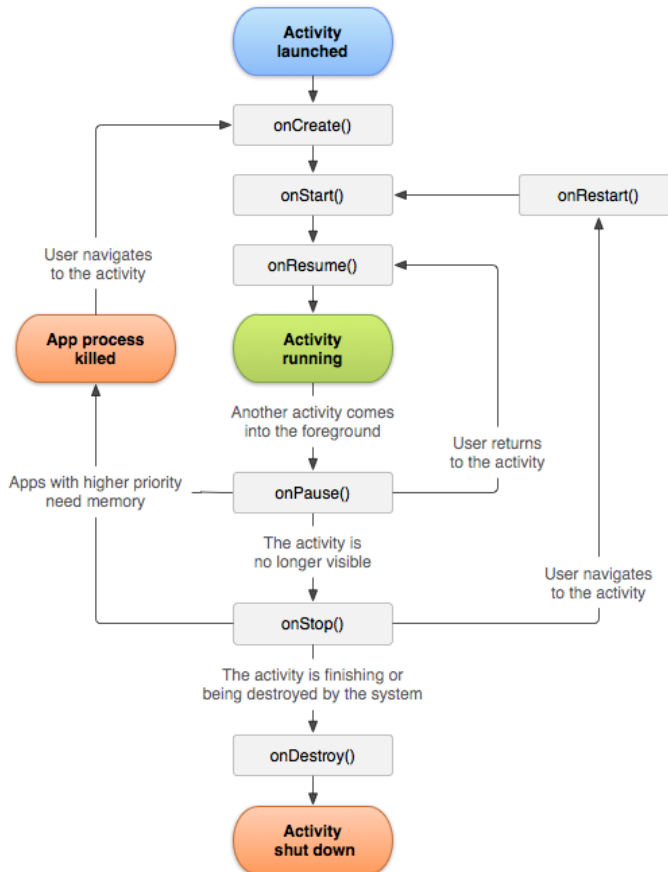
Kondisi – kondisi diatas bisa bekerja jika menggunakan *method-method* yang disediakan oleh Android [10], seperti:

- 1) *onCreate()*
Dipanggil saat aktivitas pertama kali dibuat. Di sinilah Kita harus melakukan semua persiapan pada tampilan utama. Yang selanjutnya akan menjalankan *method* `onStart()`.
- 2) *onRestart()*
Dipanggil setelah *Activity* dihentikan, hingga nantinya di panggil lagi dan menjalankan *method* `onStart()`

- 3) *onStart()*
Dipanggil tepat sebelum *Activity* terlihat di layar. Diikuti dengan method *onResume()* saat *Activity* sudah muncul di layar dan method *onStop()* jika *Activity* ingin disembunyikan.
- 4) *onResume()*
Dipanggil tepat sebelum *Activity* mulai berinteraksi dengan pengguna. Pada kondisi ini, *activity* utama berada di puncak tumpukan *Activity* yang ada. Dan selalu diikuti oleh method *onPause()*.
- 5) *onPause()*
Dipanggil bila sistem akan memulai proses untuk *Activity* lain. *Method* ini biasanya digunakan untuk menerapkan (*commit*) perubahan yang tidak tersimpan pada data persisten, menghentikan animasi dan hal-hal lain yang mungkin menghabiskan *CPU*, dan sebagainya.
- 6) *onStop()*
Dipanggil bila *Activity* tidak lagi terlihat di layar. Hal ini bisa terjadi karena *Activity* sedang ditutup, atau karena adanya *activity* lain (yang baru). Diikuti oleh method *onRestart()* jika *Activity* kembali dipanggil untuk berinteraksi dengan pengguna, atau method *onDestroy()* jika *activity* benar-benar ingin dihilangkan.
- 7) *onDestroy()*
Dipanggil sebelum *activity* benar-benar ingin tutup. Inilah panggilan terakhir yang akan diterima *Activity*.

2.4.3 Android *LifeCycle*

Diagram *Android LifeCycle* dapat ditunjukkan pada Gambar 2.1.



Gambar 2.1 *Android LifeCycle*

2.5 SQLite

Android menyediakan sebuah database yang secara default sudah ada di dalam library android, yaitu SQLite. Untuk

keperluan operasi database pada smartphone atau tablet Android, SQLite sangat memadai Karena ukurannya yang kecil, cepat dan ringan dalam hal sumber daya. Karena sifatnya sebagai *embedded database*, SQLite tidak memiliki server namun bentuknya adalah library yang akan dipanggil saat program dijalankan.

Seperti halnya database pada umumnya, SQLite memiliki objek-objek seperti table, view dan index. Perintah-perintah SQL yang digunakan yaitu *SELECT*, *INSERT*, *UPDATE*, *DELETE* dan sebagainya.

SQLite diaktifkan di dalam program dengan cara meng-*import library* SQLite pada program *header*, misalnya:

```
Import android.database.sqlite.SQLiteDatabase;
Import
android.database.sqlite.SQLiteOpenHelper;
```

SQLite adalah *Database Open Source* yang tertanam ke Android. SQLite mendukung fitur *database* relasional standar seperti sintaks SQL, *transactions* dan *prepared statements*. Selain itu hanya memerlukan sedikit memori pada saat *runtime* (sekitar 250 KByte). Menggunakan SQLite di Android tidak memerlukan *setup database* atau administrasi. Kita menentukan SQL untuk bekerja dengan *database* dan *database* secara otomatis dikelola untuk kita. SQLite mendukung tipe data *TEXT* (mirip dengan *String* di *Java*), *INTEGER* (mirip dengan yang lama di *Java*) dan *REAL* (mirip dengan ganda di *Java*). Semua jenis lain harus dikonversi ke pada bidang ini sebelum menyimpannya dalam *database*. SQLite sendiri tidak memvalidasi apakah jenis ditulis ke kolom sebenarnya dari jenis yang didefinisikan, kita juga dapat menulis sebuah *integer* ke dalam kolom *string* [11].

Untuk membuat dan meng-*upgrade* database dalam aplikasi Android biasanya menggunakan *subclass*

“*SQLiteOpenHelper*”. Di kelas ini, perlu mengesampingkan *method onCreate()* untuk membuat *database* dan *onUpgrade()* untuk meng-*upgrade* *database* dalam kasus perubahan dalam skema *database*. Kedua metode menerima “*SQLiteDatabase*” objek.

SQLiteOpenHelper menyediakan *method getReadableDatabase()* dan *getWritableDatabase()* untuk mendapatkan akses ke obyek “*SQLiteDatabase*” yang memungkinkan akses *database* baik dalam membaca atau menulis modus.

Untuk *primary key* dari *database* selalu harus menggunakan *identifier* “id” karena beberapa fungsi *Android* bergantung pada standar ini

2.6 *Decision Tree*

Decision Tree adalah struktur *flowchart* yang mempunyai *tree* (pohon), dimana setiap simpul internal menandakan suatu tes atribut. Setiap cabang merepresentasikan kelas atau distribusi kelas, alur pada *decision tree* ditelusuri dari simpul ke akar ke simpul daun yang memegang prediksi kelas untuk contoh tersebut. *Decision tree* mudah untuk dikonversikan keaturan klasifikasi (*classification rule*). Konsep data dalam *decision tree* dinyatakan dalam bentuk tabel dengan atribut dan record.

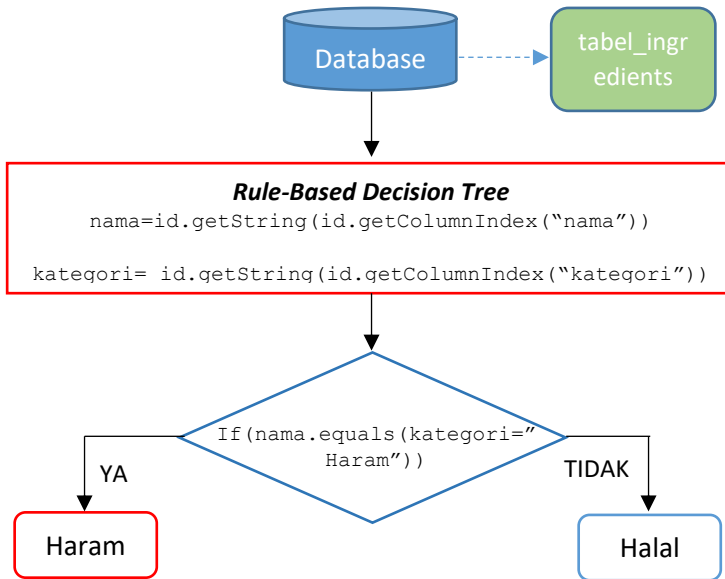
Decision Tree biasanya paling umum digunakan dalam model *Computational Intelligence* (CI). Bahkan ketika algoritma lain memberikan model yang lebih akurat (baik mendekati target), *Decision Tree* sering dianggap sebagai yang sangat menarik. Salah satu alasan yang paling penting dari daya tarik mereka adalah mudah dimengerti. *Decision Tree* dapat dengan mudah dinyatakan dalam bentuk seperangkat aturan logis yang menggambarkan fungsi keputusan. Ketika digunakan untuk mendukung keputusan, *Decision Tree* memberikan penjelasan sederhana dari keputusan tertentu,

biasanya dalam bentuk aturan logis tunggal (yang berlaku untuk kasus yang sedang dihadapi) menjadi gabungan dari beberapa tempat yang dapat dibaca [6]

2.7 Rule-Based Decision Tree

Sebuah *rule-based decision tree* digunakan sebagai sekumpulan dari rule *IF-THEN* untuk pengklasifikasian. Sebuah aturan *IF-THEN* dengan form, *IF* kondisi *THEN* hasil[12]. Contoh dari *rule IF-THEN* dapat dilihat pada Gambar 2.2.

Pada aplikasi ini, *tree* yang digunakan hanya mempunyai satu akar karena hanya mempunyai atribut *dataset ingredients* dan atribut halal dan haram. Oleh karena itu, dengan menggunakan *rule-based decision tree* ini *ingredients* yang sudah di-training sangat efektif untuk pengklasifikasian produk makanan halal dan haram



Gambar 2.2 Diagram Rule-Based Decision Tree

BAB III METODOLOGI PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang digunakan dalam penyusunan Tugas Akhir. Disamping itu, dijelaskan pula prosedur dan proses pelaksanaan tiap-tiap langkah yang dilakukan dalam menyelesaikan Tugas Akhir.

3.1 Tahapan Penelitian

Langkah-langkah yang digunakan dalam menyelesaikan Tugas Akhir ini yang diringkas dalam Gambar 3.1 adalah sebagai berikut :

1. Studi Literatur

Pada tahap ini akan dilakukan pencarian jurnal-jurnal ilmiah, pencarian buku, studi tentang software yang sudah ada sebelumnya dan sumber referensi lainnya. Mempelajari dan memahami konsep dari Android Studio, konsep dari SQLite dan DBMS, bagaimana mengimplementasikan database SQLite kedalam Android Studio untuk pembuatan *database* Android, penyimpanan *ingredients dataset* kedalam *database* menggunakan *database* SQLite, dan mengimplementasikan algoritma *decision tree* pada dataset *ingredients* untuk menghasilkan keputusan kehalalan suatu produk makanan dan minuman.

2. Pengumpulan Data

Pengumpulan Data merupakan tahap untuk mengumpulkan data yang diperlukan dalam pengerjaan Tugas Akhir ini yaitu data *ingredients* bahan makanan dan minuman. Data yang terkumpul akan digunakan sebagai masukan (input). Adapun variable data yang akan dipakai berdasarkan data *ingredients* yang dirumuskan oleh *European Economic Community* (EEC) dan diadopsi secara universal oleh industri makanan di seluruh dunia.

3. Analisis dan Perancangan Sistem

Pada tahapan ini akan dilakukan perancangan terhadap database *ingredients* untuk tempat penyimpanan data set *ingredients*. Data set ini nanti yang akan digunakan sistem untuk melakukan proses *decision tree* untuk menghasilkan *output* halal atau haram

4. Implementasi Sistem

Tahap implementasi ini terdiri dari beberapa tahap yaitu:

a) Konversi data input *ingredients* dalam *array*

Tahap ini dimulai ketika *user* menginputkan *ingredients* apa saja yang akan diperiksa. *User* dapat menginputkan banyak *ingredients* yang akan diperiksa. Lalu dari tiap *ingredients* yang diinputkan akan dikonversi menjadi *Array* agar dapat di cek satu per satu *ingredients* yang dimasukkan

b) Pendefinisian *Boolean*

Pada tahap ini adalah pendefinisian untuk *true* = 1 dan *false* = 0. Pendefinisian ini digunakan saat sistem melakukan *looping decision tree* untuk menemukan *ingredients* haram yang terdapat pada *array* inputan *user*.

c) Pengecekan *ingredients* halal dan haram pada *array*

Pada tahap ini dilakukan pengecekan pada tiap *array ingredients* input dari *user* apakah *ingredients* yang dimasukkan termasuk halal atau haram. *Array ingredients* yang diinputkan akan dicek dengan mencocokkan data dari database, lalu mengambil index kategori dari database *ingredients* tersebut untuk mendapatkan kategori halal atau haram. Jika hasil cek pertama berupa halal, maka sistem akan melakukan looping untuk *array* yang selanjutnya. Dan jika hasil cek berupa haram maka sistem akan mengeluarkan *output* berupa keputusan produk tersebut adalah haram.

5. Pengujian

Pada tahap ini akan dilakukan pengujian terhadap aplikasi dengan mencoba beberapa macam *ingredients* dari suatu produk untuk mengetahui apakah suatu produk tersebut halal atau haram. Kemudian pengujian terhadap kesesuaian *keyword* dengan *ingredients dataset* dan kecepatan aplikasi dalam menyimpulkan hasil serta perbandingan aplikasi dengan aplikasi yang telah ada sebelumnya.

6. Kesimpulan dan Saran

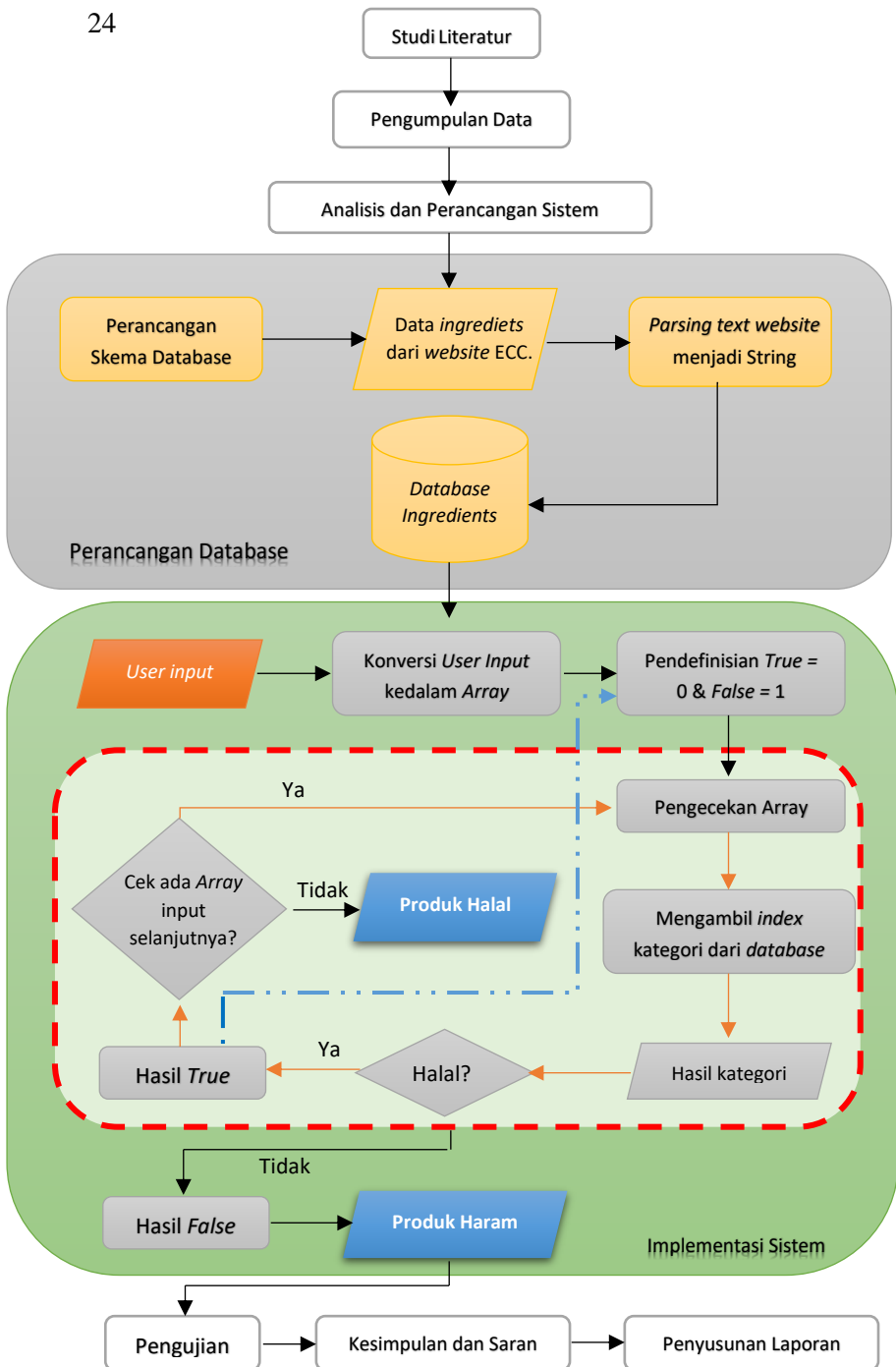
Setelah dilakukan analisis dan pembahasan maka dapat ditarik suatu kesimpulan dan saran sebagai masukan untuk pengembangan penelitian lebih lanjut.

7. Penyusunan Laporan Tugas Akhir

Setelah semua proses selesai dilakukan maka tahap terakhir adalah penyusunan laporan Tugas Akhir.

3.2 Diagram Alir Penelitian

Berdasarkan uraian tersebut diatas, penelitian Tugas Akhir ini dapat dinyatakan dalam diagram alir sebagai berikut.



Gambar 3.1 Diagram Alir Penelitian

BAB IV

PERANCANGAN APLIKASI DAN IMPLEMENTASI

Pada bab ini akan dijelaskan perancangan aplikasi dan implementasi sistem yang akan dikembangkan. Perancangan aplikasi ini diwujudkan kedalam sebuah aplikasi sistem *mobile phone* yang berbasis android dengan menggunakan *software* Android Studio dan XML untuk tampilan *interface*-nya. Aplikasi ini digunakan untuk memberikan informasi kandungan haram dalam produk makanan dan minuman.

4.1 Perancangan Aplikasi

4.1.1 Perancangan Database

Perancangan database merupakan hal yang sangat penting dalam membangun sebuah aplikasi. Dalam aplikasi ini database digunakan untuk menyimpan data *ingredients*, jenis *ingredients* dan kategori *ingredients* itu sendiri. Dalam database aplikasi ini hanya digunakan 2 tabel yaitu tabel *ingredients* dan tabel *hasil*. Dimana kedua tabel tersebut dihubungkan oleh sebuah *table relasi*. Database ini menggunakan database *SQLite*. Perancangan *database* pada aplikasi ini dapat dilihat pada Gambar 4.1.

Keterangan dari struktur masing-masing tabel akan dijelaskan sebagai berikut:

1. Tabel *ingredients*

Tabel *ingredients* adalah tabel yang berisi data *ingredients* produk makanan.

No	Atribut	Tipe Data	Keterangan
1	id	Integer	Id Auto Increment
2	id_ingredients	Varchar	Kode ingredients

3	nama	Text	Nama ingredients
4	kategori	Text	Kategori ingredients
5	keterangan	Text	Keterangan

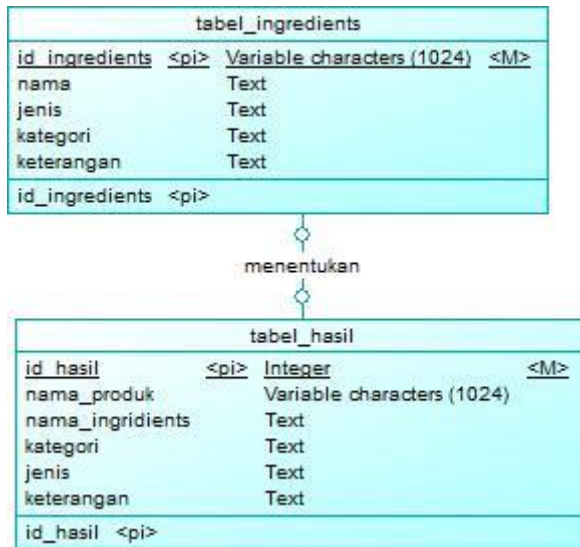
2. Tabel tabel_hasil

Tabel tabel_hasil adalah tabel yang berisi data *ingredients* apa saja yang sudah dicek oleh *user* dan juga untuk penyimpanan hasil cek dari *user*.

No	Atribut	Tipe Data	Keterangan
1	id	Integer	Id Auto Increment
2	nama_produk	Varchar	Nama produk
3	nama_ingredients	Text	Nama ingredients
4	kode	Text	Kode ingredients
5	jenis	Text	Jenis ingredients
6	keterangan	Text	Keterangan

4.1.2 Perancangan Sistem

Sistem ini dirancang dengan menggunakan *software* Android Studio dan diimplementasikan menjadi aplikasi android. Perancangan sistem ini dimulai dari pembuatan *interface* android dengan menggunakan XML Android Studio, perancangan koneksi database, pengambilan data dari website, *parsing* data dari website kedalam *String*, pembuatan algoritma *decision tree*, pengimplementasian algoritma *decision tree* pada data set *ingredients*, dan pembuatan hasil keputusan sistem.



Gambar 4.1 CDM pada Aplikasi Android

4.2 Analisis Kebutuhan Sistem

Saat ini banyak sekali produk makanan dan minuman yang beredar di Indonesia. Ada produk yang sudah berlabel halal dan juga masih ada yang belum berlabel halal. Menurut data yang didapat pada tahun 2015 ada 816 produk dan 11 suplemen makanan yang terdaftar di BPOM dan diantaranya masih terdapat di BPOM dan diantaranya juga masih terdapat produk non-halal. Dan pada tahun 2017 saat ini, produk makanan yang digemari remaja yaitu Samyang telah terbukti terindikasi babi. Padahal makanan khas Korea tersebut masuk ke Indonesia melalui uji sertifikasi dari BPOM.

Permasalahan diatas dapat diatasi jika terdapat pengecekan terhadap *ingredients* yang tercantum pada produk makanan tersebut. Oleh karena itu dibuat aplikasi android yang dapat digunakan semua orang untuk memeriksa

ingredients yang tercantum pada produk makanan tersebut. Dengan menggunakan metode *decision tree*, akan sangat mudah untuk mengkalsifikasikan produk makanan tersebut halal atau haram melalui *ingredients* yang dimasukkan.

4.3 Analisis Kebutuhan User

Di era teknologi saat ini, masyarakat banyak sekali yang menggunakan *mobile phone* untuk keperluan sehari-hari. Dan kebanyakan dari mereka menggunakan sistem operasi Android pada *mobile phone* mereka.

Melihat dari banyaknya kasus makanan berlabel halal tetapi masih mengandung bahan makanan haram didalamnya, masyarakat yang terutama muslim perluantisipasi dengan produk makanan tersebut. Untuk memudahkan masyarakat dibuatlah aplikasi ini untuk memeriksa apakah produk tersebut halal atau haram dengan memasukkan *ingredients* dari produk tersebut.

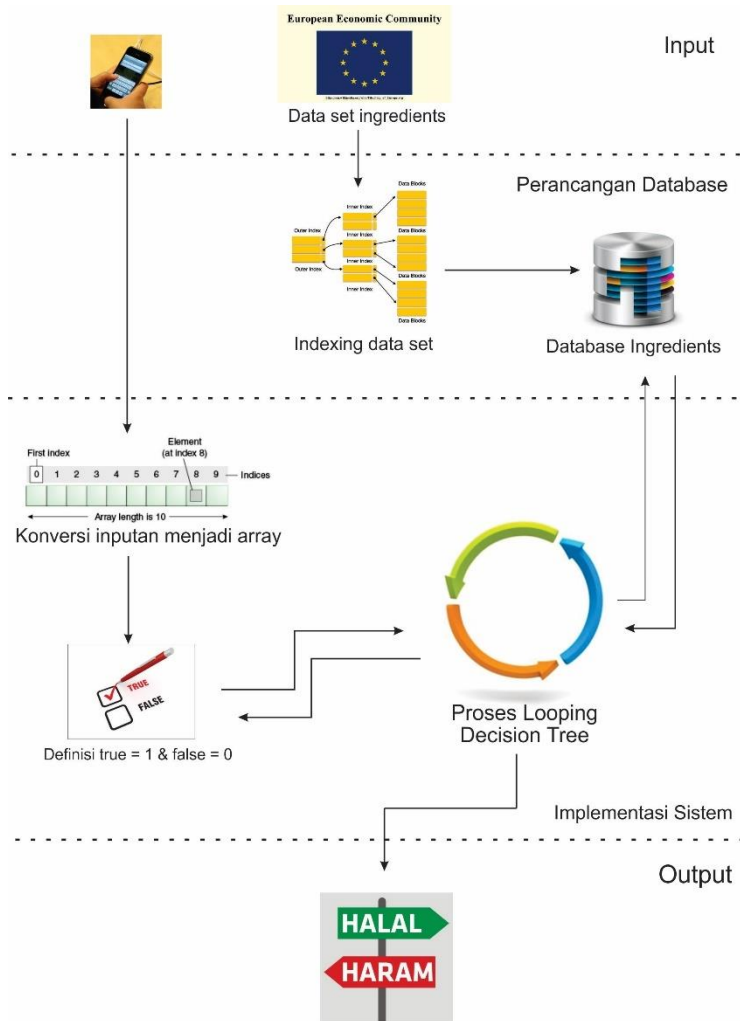
Aplikasi ini sangat *user friendly* atau sangat mudah dioperasikan oleh masyarakat dengan tampilan yang mudah dipahami. *User* hanya perlu memasukkan nama produk dan *ingredients* apa saja yang terdapat dalam produk makanan yang ingin diperiksa. Lalu sistem akan memproses *ingredients* yang akan diinputkan dan akan menghasilkan keputusan produk tersebut halal atau haram.

4.4 Desain Sistem

Pada tugas akhir ini data yang digunakan adalah data set *ingredients* yang didapat dari ECC. Data ini berisi kode *ingredients*, nama *ingredients*, kategori *ingredients*, dan keterangan *ingredients*. Data tersebut terdapat dalam website ECC, jadi data tersebut perlu di *parsing* menjadi sebuah input yang bisa terbaca oleh database SQLite android. Setelah data

tersebut di *parsing* menjadi *String* maka data tersebut akan disimpan didalam database.

Setelah data tersebut disimpan dalam database, data set tersebut nantinya digunakan dalam proses *looping decision tree* untuk menentukan *ingredients* yang dimasukkan oleh *user*. *Ingredients* yang dimasukkan oleh *user* akan dikonversi menjadi *array* terlebih dahulu agar dapat di cek satu per satu *ingredients* yang diinputkan. Setelah dikonversi menjadi *array* sistem akan mendefinisikan *Boolean true* = 1 dan *false* = 0. Pendefinisian ini digunakan untuk menentukan *ingredients* yang diinputkan halal atau haram. Setelah melakukan pendefinisian *Boolean*, aplikasi akan melakukan looping pada setiap *array* yang diinputkan user untuk memeriksa setiap atribut kategori *ingredients* tersebut. Ketika *array* yang diperiksa menghasilkan *true* maka sistem akan terus *looping* untuk mencari *ingredients* haram. Ketika *array* sudah dicek semua hasil *true* maka akan menghasilkan produk halal, sebaliknya jika sistem menghasilkan hasil *false* maka menghasilkan *output* haram.

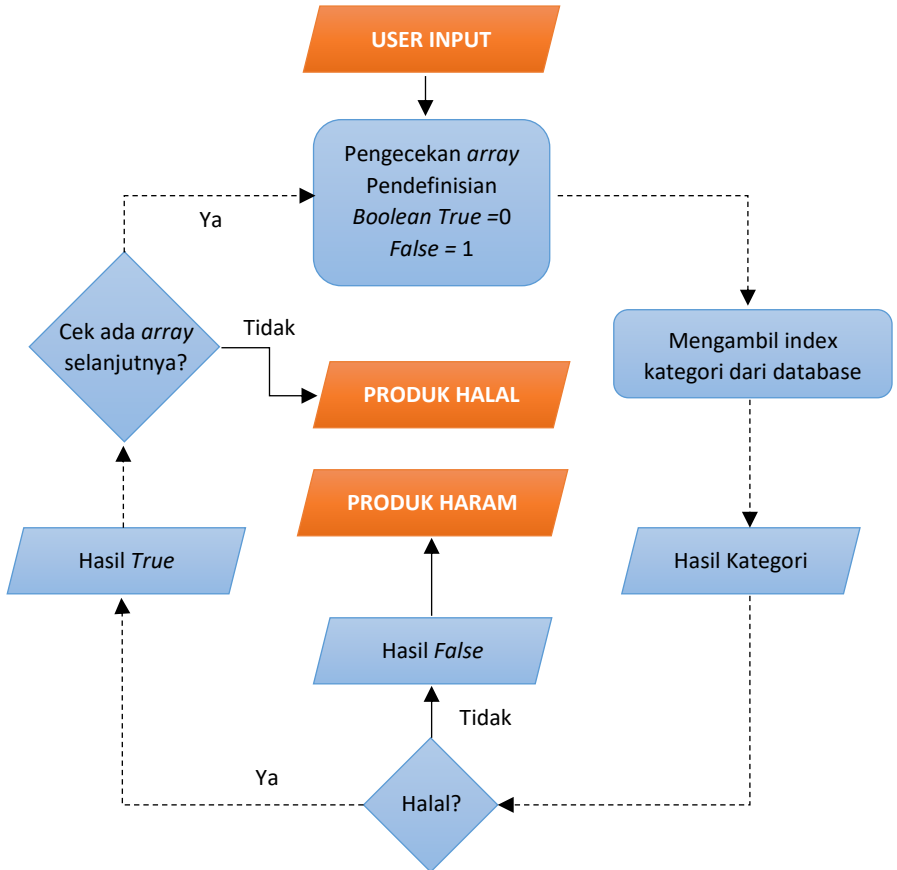


Gambar 4.2 Desain Sistem Perangkat Lunak

4.5 Desain Proses *Decision Tree*

4.5.1 Desain *Decision Tree* Pada Aplikasi

Berikut adalah desain *decision tree* yang diterapkan pada aplikasi ini.



Gambar 4.3 Desain *Decision Tree*

Proses *decision tree* pertama kali dilakukan dengan memeriksa masukkan *ingredients* dari *user* yang berbentuk *array* dan pendefinisian *Boolean true* dan *false*. Pendefinisian ini digunakan untuk memeriksa index kategori *array ingredients* apakah *ingredients* tersebut halal atau haram. Setelah dilakukan pengecekan *array* sistem akan memeriksa *array ingredients* tersebut dengan mencocokkan dari database *ingredients*. Lalu sistem akan mengambil nilai index kategori dari database *ingredients* yang berisi keterangan halal atau haram. Tahap selanjutnya sistem akan memeriksa hasil pengambilan nilai index kategori tersebut halal atau haram. Ketika hasilnya adalah halal, maka sistem akan mendefinisikan *array ingredients* tersebut sebagai *true*. Setelah didefinisikan *true* lalu sistem akan memeriksa apakah ada *array ingredients* selanjutnya. Ketika sistem mendapati ada *array* selanjutnya maka sistem akan melakukan *looping* dengan menjalankan pengecekan *array* selanjutnya.

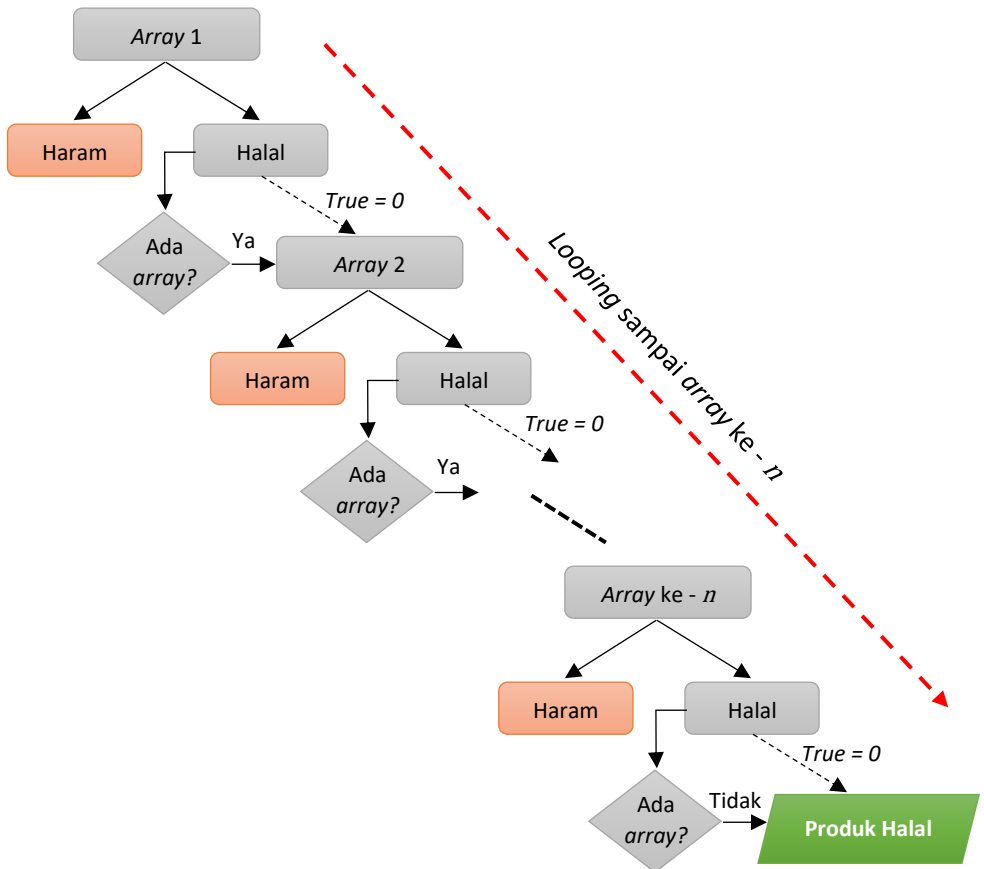
Jika hasil pengambilan nilai index kategori bernilai haram maka sistem akan mendefinisikan *array ingredients* tersebut sebagai *false* dan langsung akan menghasilkan *output* produk tersebut adalah haram.

4.5.1 Desain Tree

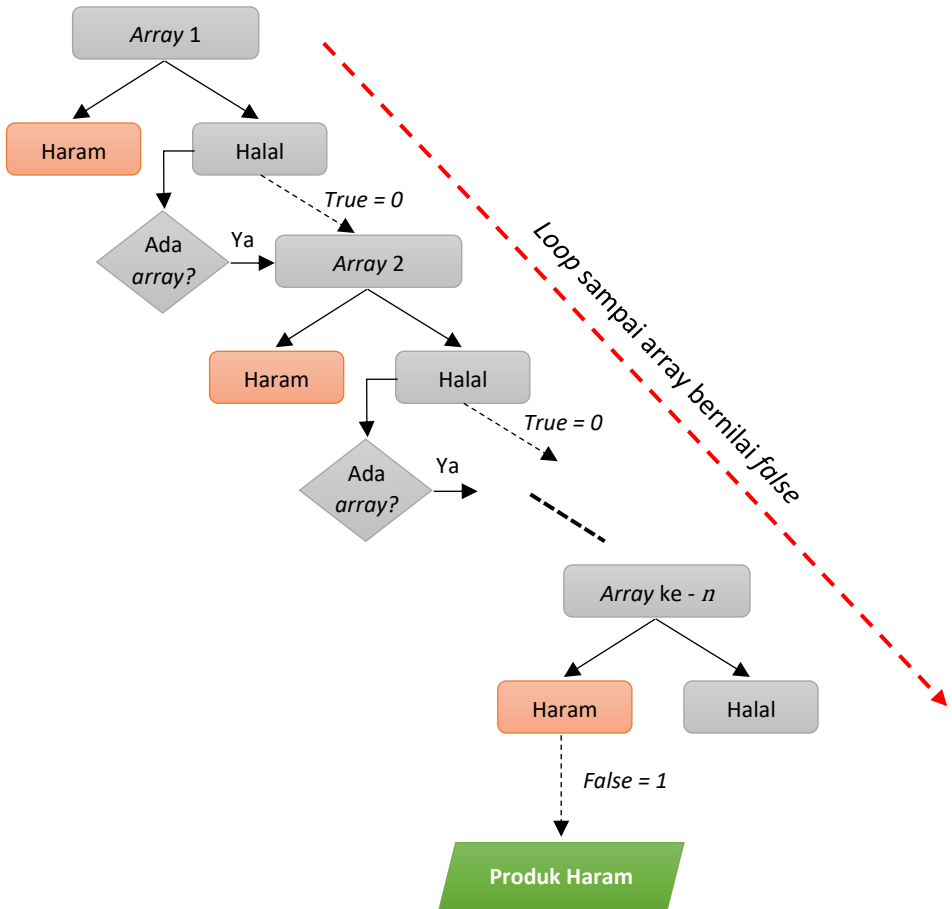
Berikut adalah desain *tree* yang digunakan dalam *decision tree*, dapat dilihat pada gambar 4.4 dan 4.5. Pada saat *ingredients* yang diinputkan oleh *user* adalah halal, maka sistem akan mendefinisikan *ingredients* tersebut sebagai *true* lalu memeriksa semua *array ingredients* yang diinputkan oleh *user* dan melakukan *looping* sampai semua *ingredients* diperiksa.

Akan tetapi jika *user* menginputkan *ingredients* yang halal lalu ada satu saja *ingredients* yang haram. Maka sistem

akan *looping* hanya sampai *array* yang diinputkan bernilai *false* dan akan menghasilkan *output* produk haram.



Gambar 4.4 Desain *Tree* saat produk halal



Gambar 4.5 Desain *Tree* saat produk hram

4.6 Analisis Aplikasi

4.6.1 Menentukan Jenis Data Masukan

Dalam penentuan jenis data yang dimasukkan, penulis menggunakan data yang didapat dari *European Economic Community (E-Numbers)*. Data yang digunakan adalah data *ingredients* bahan makanan, meliputi jenis *ingredients*, kategori *ingredients*, dan keterangan *ingredients* itu sendiri. Data ini juga yang nantinya akan digunakan *user* untuk memasukkan nama *ingredients* kedalam aplikasi untuk menentukan suatu produk makanan tersebut halal atau haram.

4.6.2 Jenis Output

Jenis *output* yang dihasilkan oleh aplikasi ini adalah berupa keputusan halal atau haram. Setelah *user* memasukkan daftar nama *ingredients* yang ingin dicek, maka aplikasi akan mengolah *input user* dan diproses menggunakan *decision tree* yang akan menghasilkan *output* keputusan berupa *pop-up* halal atau haram.

4.6.3 Analisis Komponen Sistem

4.6.3.1 Use Case Diagram Pengguna

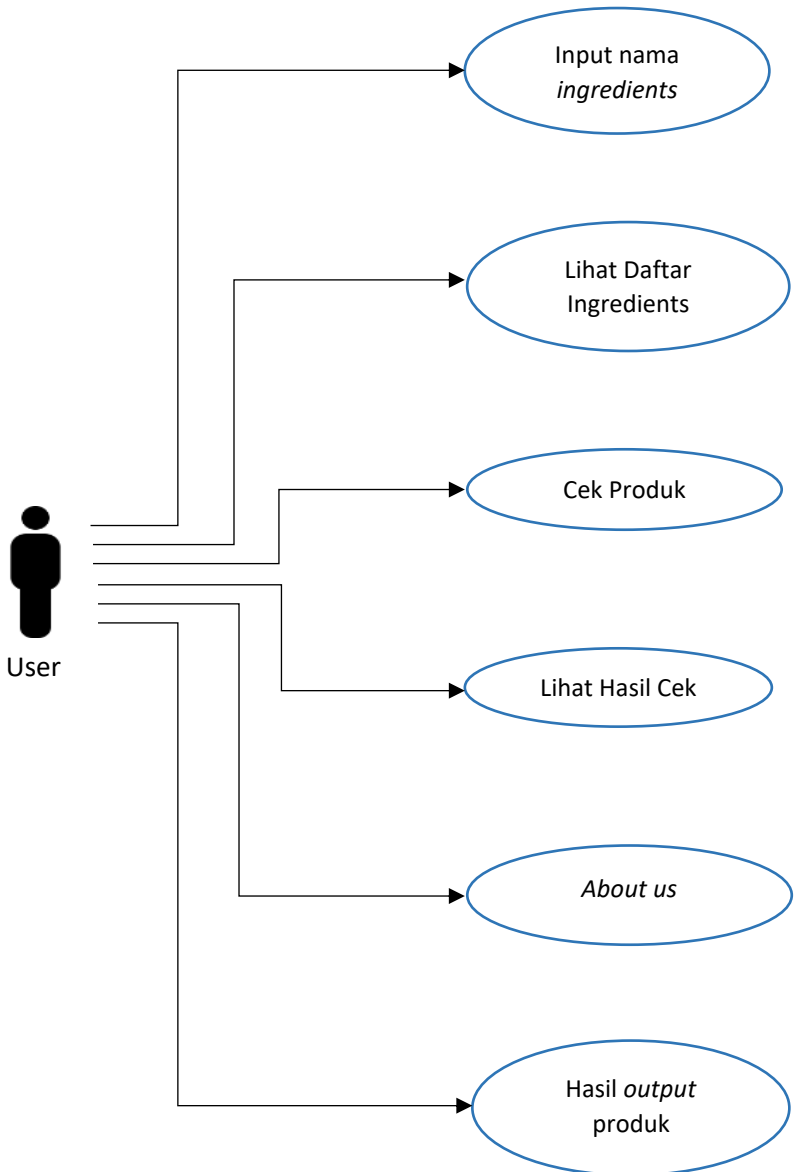
Use case diagram pengguna merupakan gambaran skenario dari interaksi yang terjadi antara pengguna dan sistem. *Use case diagram* menggambarkan suatu hubungan interaksi antar aktor(pengguna) terhadap sistem aplikasi. *Use case diagram* pengguna dapat dilihat pada Gambar 4.2.

Pada aplikasi ini tidak ada menu login, jadi aplikasi ini murni untuk *user*. Pertama kali *user* masuk akan dihadapkan pada *menu Home*. *Menu home* berisi pilihan cek produk, tambah *ingredients*, lihat daftar *ingredients*, dan lihat hasil cek produk. Pilihan cek produk berisi tampilan untuk cek produk makanan atau minuman dengan memasukkan daftar *ingredients*-nya. Pilihan tambah *ingredients* berisi tampilan

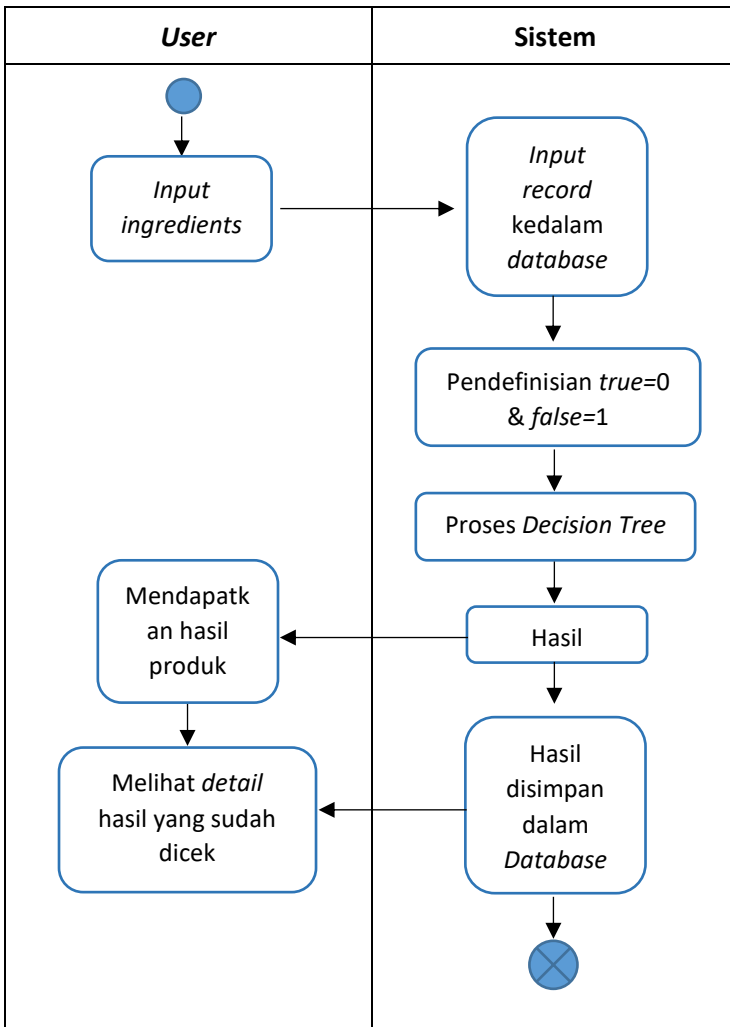
untuk menambahkan daftar *ingredients* yang belum ada pada *database*. Pilihan daftar *ngredients* berisi tampilan untuk melihat semua daftar *ingredients* pada *database*. Dan terakhir pilihan lihat hasil cek produk berisi tampilan untuk melihat hasil produk yang sudah di cek beserta *ingredients* yang tertera pada produk tersebut.

4.6.3.2 Activity Case Diagram Sistem

Activity case diagram sistem berisi tentang gambaran bagaimana sistem pada aplikasi ini bekerja. *Activity case diagram* sistem bisa dilihat pada Gambar 4.3.



Gambar 4.6 Use Case Diagram User

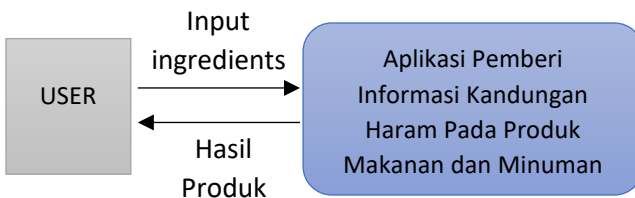


Gambar 4.7 Activity Diagram

4.6.3.3 Data Flow Diagram

Data flow diagram adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). Kemudian dibuat suatu *data flow diagram* untuk menggambarkan alur proses. DFD *Level 0* biasa disebut dengan diagram sistem inti. Pada diagram sistem ini menjelaskan sebuah proses umum yang nantinya akan di dekomposisi menjadi proses-proses yang lebih detail. Seperti gambar 4.8 berikut

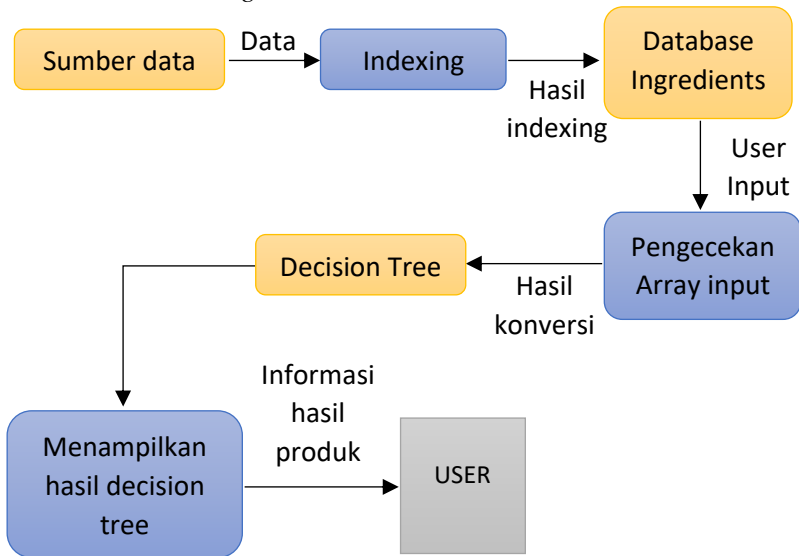
1. Data Flow Diagram Level 0



Gambar 4.8 Data Flow Diagram Level 0

Proses utama pada sistem ini digambarkan pada gambar diatas. Sebagai input, *user* memasukkan *ingredients* yang akan diproses pada aplikasi ini dan sebagai *output* dari aplikasi ini *user* akan mendapatkan informasi berupa produk yang di cek halal atau haram.

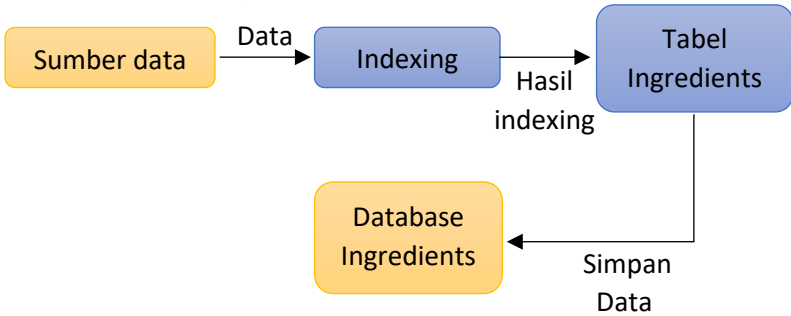
2. *Data Flow Diagram Level 1*



Gambar 4.9 *Data Flow Diagram Level 1*

Proses selanjutnya adalah proses yang dilakukan oleh aplikasi ini seperti gambar diatas. Proses ini dimulai dengan *indexing* dari sumber data. Data yang ada dalam *database ingredients* akan digunakan untuk proses *decision tree*. Setelah melakukan *decision tree* maka sistem akan menampilkan hasil *decision tree* pada *user*.

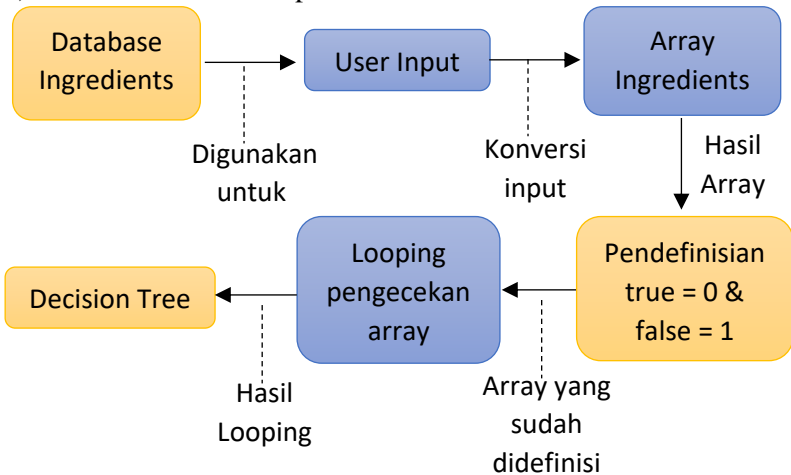
3. Data Flow Diagram Level 2



Gambar 4.10 Data Flow Diagram Level 2

Pada DFD *level 2* ini yaitu proses penyimpanan data kedalam *database ingredients*. Data yang sudah didapat akan dilakukan *indexing*. Setelah disimpan dalam tabel *ingredients* maka sistem akan menyimpan pada *database ingredients*.

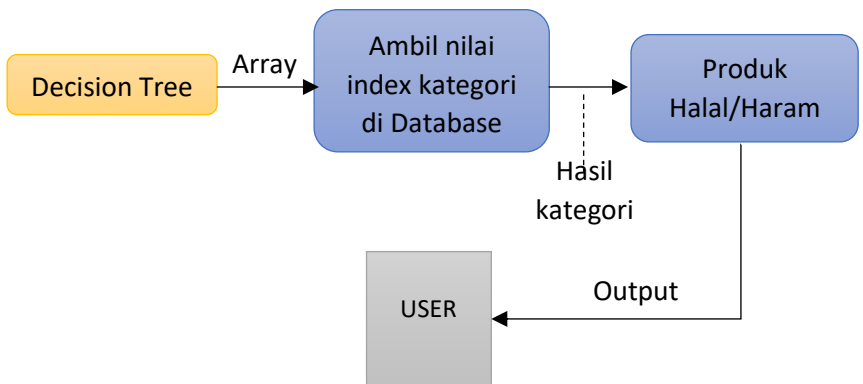
a) DFD Level 2 untuk proses *decision tree*



Gambar 4.11 DFD Level 2 untuk proses *decision tree*

Pada proses ini *database ingredients* yang sudah dibuat akan digunakan *user* untuk menginputkan *ingredients* produk yang ingin diperiksa. Lalu *ingredients* yang diinputkan dari *user* akan dikonversi menjadi *array* agar dapat diperiksa *ingredients*-nya satu per satu. Setelah dikonversi menjadi *array* akan dilakukan pendefinisian Boolean *true* = 0 & *false* = 1. Setelah melakukan pendefinisian sistem akan melakukan *looping* untuk pengecekan tiap *array*. Lalu hasil *looping* tadi akan dilakukan *decision tree*.

- b) DFD *level 2* untuk menampilkan hasil produk halal/haram



Gambar 4.12 DFD *Level 2* untuk menampilkan hasil produk halal/haram

Pada proses ini data yang sudah didefinisi tadi akan dilakukan *decision tree*. Langkah pertama adalah *array ingredients* yang diinputkan akan mengambil nilai index kategori dalam *database ingredients*. Setelah mengambil hasil kategori maka sistem akan mendefinisikan *array* tersebut *true* atau *false*. Ketika sistem hanya menemukan

ingredients halal dalam *array ingredients* maka sistem akan menampilkan produk halal. Tetapi jika sistem menemukan ada *ingredients* haram maka produk akan mengeluarkan hasil produk haram pada *user*.

4.7 Perancangan User Interface

User interface adalah aspek sistem computer yang dapat dilihat oleh pengguna sistem. Perintah-perintah dan mekanisme yang digunakan pengguna untuk mengoperasikan sistem. Berikut ini perancangan desain *user interface* aplikasi:

4.7.1 Menu Halaman Utama

Perancangan form menu halaman utama ini merupakan tampilan menu yang pertama kali muncul saat *user* menggunakan aplikasi ini. Pada halaman ini akan ditampilkan 4 menu utama pada aplikasi ini, yaitu cek produk, tambah *ingredients*, daftar *ingredients*, lihat hasil cek produk.

4.7.2 Menu Cek Produk

Menu ini adalah menu utama dalam aplikasi ini, pada menu ini *user* diminta untuk memasukkan nama produk yang ingin dicek dan memasukkan *ingredients* apa saja yang ingin dicek.

4.7.3 Menu Daftar *Ingredients*

Pada menu ini berisi semua daftar *list ingredients* yang ada dalam database yang direpresentasikan oleh gabungan *RecyclerView* dan *CardView*. Pada menu ini *user* dapat melihat dan mencari daftar *ingredients* apa saja yang terdapat pada aplikasi ini.

4.7.4 Menu Lihat Hasil Cek Produk

Menu ini berisi hasil produk yang sudah dicek oleh *user*. Data hasil cek produk disajikan dalam bentuk *ListView* dengan nama produknya. Dan *ListView* tersebut dapat di klik untuk

melihat *ingredients* apa saja yang terkandung dalam produk tersebut.

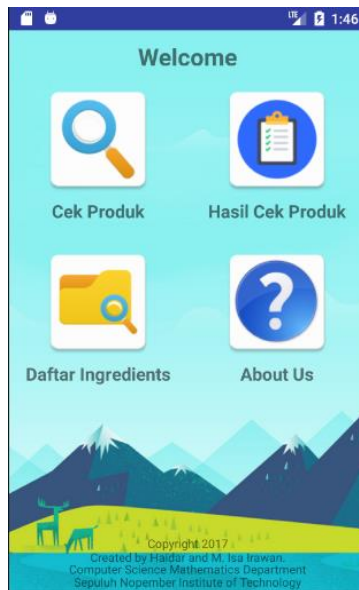
BAB V

UJI COBA DAN PEMBAHASAN

Pada bab ini dijelaskan tentang implementasi dalam bahasa pemrograman Java dan hasil uji coba program menggunakan *mobile phone* berbasis android.

5.1 Pengujian Halaman Utama

Pada tahap ini akan dilakukan pengujian pada halaman awal program ini. Pada halaman awal terdapat empat pilihan yaitu cek produk, tambah ingredients, daftar ingredients, hasil cek produk. Hal tersebut ditunjukkan pada Gambar 5.1.



Gambar 5.1 Tampilan Awal

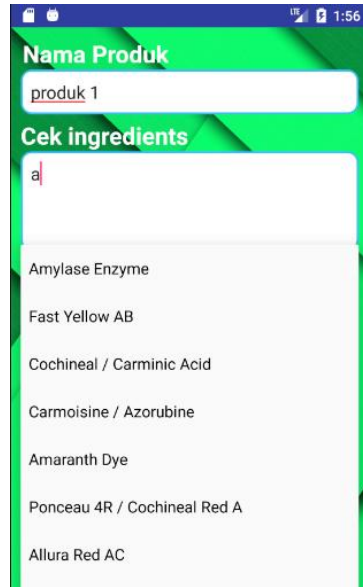
Pada halaman utama, *user* akan langsung dihadapkan pada empat pilihan utama. *User* bebas untuk memilih menu apa saja yang tertera pada halaman utama. Menu cek produk adalah menu yang digunakan *user* untuk cek produk makanan dan minuman. Menu daftar ingredients adalah menu untuk melihat daftar *ingredients* yang ada pada *database* aplikasi. Lalu menu *About Us* adalah menu untuk melihat informasi aplikasi ini. Dan untuk menu hasil cek produk adalah untuk melihat lebih rinci produk makanan atau minuman yang sudah di cek kehalalannya.

5.2 Pengujian Menu Cek Produk

Menu cek produk ini adalah program utama dalam aplikasi ini. Didalam menu ini berisi dua buah isian yaitu nama produk dan *ingredients* yang ingin di cek, seperti yang ditunjukkan pada Gambar 5.2. Untuk isian *ingredients*, user tidak perlu ketik keseluruhan nama *ingredients*-nya, karena *TextField* untuk isian *ingredients* berbentuk *AutoComplete*, seperti yang ditunjukkan pada Gambar 5.3. Jadi *user* hanya perlu mengetik satu atau dua huruf maka saran *ingredients* akan keluar, jadi *user* hanya perlu memilih *ingredients*-nya.



Gambar 5.2 Menu
Cek Produk



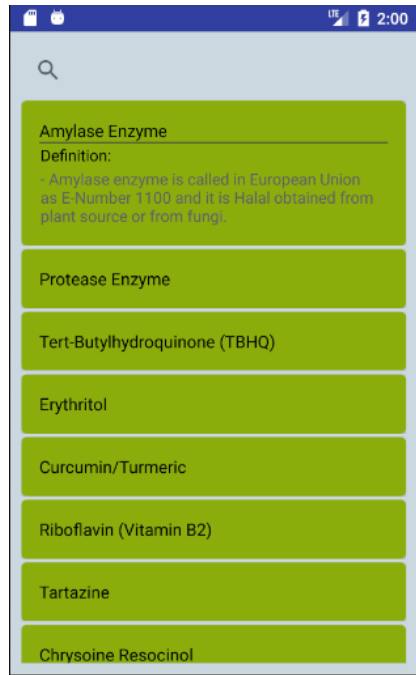
Gambar 5.3 TextField
AutoComplete

5.3 Pengujian Daftar Ingredients

Menu ini berfungsi untuk melihat dan mencari ingredients yang ada dalam database aplikasi. *List ingredients* dibuat menggunakan *RecyclerView* yang dapat di-*expand* untuk melihat keterangan dari *ingredients* tersebut. Menu tambah *ingredients* ditunjukkan pada Gambar 5.4.



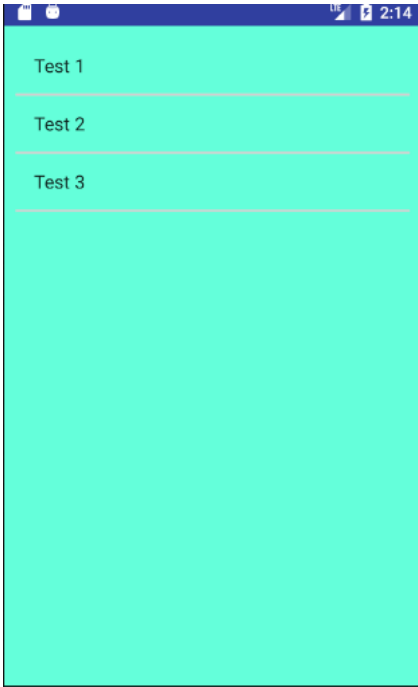
Gambar 5.4 Menu daftar *ingredients*



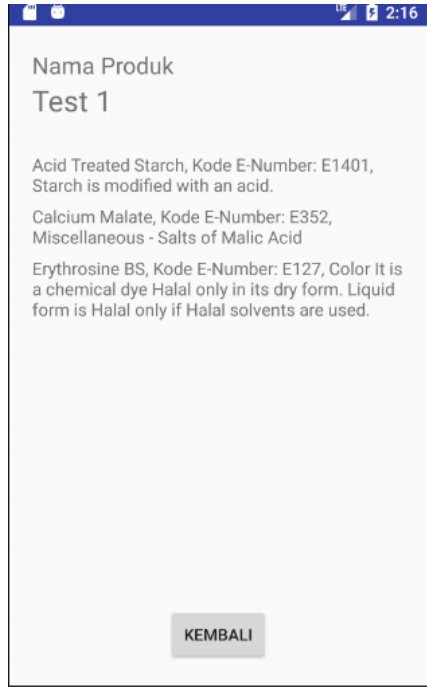
Gambar 5.5 Menu daftar *ingredients* setelah di klik

5.4 Pengujian Menu Lihat Hasil Cek Produk

Menu ini berfungsi untuk melihat hasil rincian produk yang *diinput* oleh *user*. Menu ini berisi nama produk serta *ingredients*-nya yang *diinputkan*. Sistem akan memunculkan nama beserta keterangannya dalam bentuk *ListView*. Untuk menu lihat hasil cek dapat dilihat pada Gambar 5.6, sedangkan untuk tampilan hasil setelah di klik dapat dilihat pada Gambar 5.7.



Gambar 5.6
Tampilan List Hasil



Gambar 5.7
Tampilan Hasil
Setelah di klik

5.5 Pengujian Proses *Decision Tree*

Pada bagian ini akan dilakukan pengujian pada aplikasi ini dengan menggunakan *decision tree*. Setelah *user* meng-*input*-kan nama produk maka sistem akan melakukan proses *decision tree* yang akan menghasilkan *pop-up* yang memberikan informasi produk tersebut halal atau haram. Untuk penjelasannya bisa dilihat pada Gambar 5.9. dan Gambar 5.10



Gambar 5.9
Pengujian aplikasi



Gambar 5.10
Proses *Decision*

BAB VI

PENUTUP

Pada bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan. dan saran yang dapat digunakan jika penelitian ini dikembangkan.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian program, maka dapat diambil kesimpulan sebagai berikut :

1. Algoritma *Decision Tree* telah berhasil diterapkan untuk klasifikasi *ingredients* dan memberikan informasi berupa kehalalan suatu produk dengan menggunakan data yang dikeluarkan oleh *ECC (European Economic Community)*.
2. Aplikasi ini dapat menyediakan informasi tentang macam-macam *ingredients* yang mempunyai kandungan haram dan keterangannya secara rinci.
3. Aplikasi ini dapat digunakan untuk seluruh produk makanan dan minuman buatan pabrik, karena pengecekannya menggunakan input *ingredients*.

6.2 Saran

Ada beberapa hal yang penulis sarankan untuk pengembangan penelitian selanjutnya :

1. Perlu adanya penelitian dengan menggunakan metode lain, sehingga dapat digunakan sebagai pembanding untuk mendapatkan hasil diagnose yang lebih baik
2. Agar pengguna dapat semakin memahami informasi yang ada, maka perlu ditambahkan pula gambar dari masing masing *ingredients*.

“Halaman ini sengaja dikosongkan”

DAFTAR PUSTAKA

- [1] Indonesia, Badan Pusat Statistika. (2015). <http://www.bps.go.id/> [Diakses pada tanggal 13 Februari 2017]
- [2] Indonesia, Lembaga Pengkajian Pangan Obat-obatan dan Kosmetika Majelis Ulama Indonesia (LPPOM MUI). (2016). <http://halalmui.org/> [Diakses pada tanggal 13 Februari 2017]
- [3] <http://www.indonesia-investments.com/id/budaya/agama/item69?> [Diakses pada tanggal 13 Februari 2017]
- [4] Abdul Aziz, Y. & Vui, C.N. (2012). *The role of Halal awareness and Halal certification in influencing non-Muslim's purchasing intention. Paper presented at 3rd International Conference on Business and Economic Research (3rd ICBER 2012) Proceeding*, 1819-1830.
- [5] Yunuz M., Rashid W., Ariffin M., & Rasyid M. (2014). *Muslim's Purchase Intention Towards Non-Muslim's Halal Packaged Food Manufacturer*.
- [6] Grąbczewski, Krzysztof. (2014). *Meta-Learning in Decision Tree Induction*.
- [7] Athoillah, M dan Irawan, M.I. (2013). *Perancangan Sistem Informasi Mobile Berbasis Android Untuk Kontrol Persediaan Barang di Gudang*. Institut Teknologi Sepuluh Nopember. Surabaya, Indonesia.
- [8] Bintang, R dan Irawan, M.I. (2015). *Perancangan Prototipe Sistem Pakar Berbasis Android untuk Pendukung Keputusan Diagnosa Kerusakan Pada Mobil*

di Auto2000. Institut Teknologi Sepuluh Nopember. Surabaya, Indonesia.

- [9] <https://special.worldofislam.info/Food/numbers.html> [Diakses pada tanggal 21 Februari 2017]
- [10] <https://developer.android.com> [Diakses pada tanggal 31 juli 2017]
- [11] <https://id.wikipedia.org/wiki/SQLite> [Diakses pada tanggal 31 juli 2017]
- [12] Fanguy, R dan Raghavan, V. *Generating Rule-Based Trees from Decision Trees for Concept-Based Information Retrieval*.

LAMPIRAN

MainActivity.java

```

public class MainActivity extends
AppCompatActivity {

    private DatabaseHelper dbHelper;
    SQLiteDatabase sqLiteDatabase;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if (android.os.Build.VERSION.SDK_INT >
9) {
            StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().bu
ild();
            StrictMode.setThreadPolicy(policy);
        }

        dbHelper = new DatabaseHelper(this);
        sqLiteDatabase =
dbHelper.getWritableDatabase();

        Button btnUtama =
(Button)findViewById(R.id.btnUtama);
        btnUtama.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new
Intent(getApplicationContext(), SearchUtama.class
);
                startActivity(i);
            }
        });
    }
}

```

```

        Button btnList =
        (Button) findViewById(R.id.btnList);
        btnList.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new
                Intent(getApplicationContext(), ListIngredients.class);
                startActivity(i);
            }
        });
        Button btnList2 =
        (Button) findViewById(R.id.btnList2);
        btnList2.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new
                Intent(getApplicationContext(), AddList.class);
                startActivity(i);
            }
        });

        Button btnHasil =
        (Button) findViewById(R.id.btnHasil);
        btnHasil.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new
                Intent(getApplicationContext(), LihatHasil.class)
                ;
                startActivity(i);
            }
        });
    }
}

```

DatabaseHelper.java

```

public class DatabaseHelper extends
SQLiteOpenHelper {

    ArrayList<String> list = new
ArrayList<String>();

    private static DatabaseHelper dbHelper;
    private SQLiteDatabase database;

    private static final int DATABASE_VERSION =
2;
    private static final String DATABASE_NAME =
"db_data";

    public static final String TABLE_NAME =
"tabel_ingredients";
    public static final String TABLE_HASIL =
"tabel_hasil";

    //Buat Kolom Tabel Ingredients
    private static final String KEY_ID = "id";
    private static final String
KEY_IDINGREDIENTS = "id_ingredients";
    private static final String KEY_NAMA =
"nama";
    private static final String KEY_KATEGORI =
"kategori";
    private static final String KEY_KETERANGAN =
"keterangan";

    //Buat Kolom Tabel Hasil
    public static final String HASIL_ID = "id";
    public static final String HASIL_PRODUK =
"nama_produk";
    public static final String HASIL_INGREDIENTS
= "nama_ingredients";
    public static final String KODE_INGREDIENTS
= "kode";
    public static final String HASIL_KATEGORI =
"kategori";
    public static final String HASIL_KETERANGAN

```

```

= "keterangan";

    private Context mContext;

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null,
DATABASE_VERSION);
        this.mContext = context;
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_TABLE = "CREATE TABLE " +
TABLE_NAME + "(" +
            KEY_ID + " INTEGER PRIMARY KEY
AUTOINCREMENT," +
            KEY_IDINGREDIENTS + " VARCHAR,"
+
            KEY_NAMA + " TEXT," +
            KEY_KATEGORI + " TEXT," +
            KEY_KETERANGAN + " TEXT" + ")";

        String CREATE_TABLE_KETERANGAN = "CREATE
TABLE " + TABLE_HASIL + "(" +
            HASIL_ID + " INTEGER PRIMARY KEY
AUTOINCREMENT," +
            HASIL_PRODUK + " VARCHAR," +
            HASIL_INGREDIENTS + " TEXT," +
            KODE_INGREDIENTS + " TEXT," +
            HASIL_KATEGORI + " TEXT," +
            HASIL_KETERANGAN + " TEXT" +
");";

        db.execSQL(CREATE_TABLE);
        db.execSQL(CREATE_TABLE_KETERANGAN);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int
oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " +
TABLE_NAME);
        db.execSQL("DROP TABLE IF EXISTS " +
TABLE_HASIL);
        onCreate(db);
    }

```

```

    }

    public void addData(String id_ingredients,
String nama, String kategori, String
keterangan){
        SQLiteDatabase db =
this.getWritableDatabase();

        ContentValues values = new
ContentValues();
        values.put(KEY_IDINGREDIENTS,
id_ingredients);
        values.put(KEY_NAMA, nama);
        values.put(KEY_KATEGORI, kategori);
        values.put(KEY_KETERANGAN, keterangan);

        db.insert(TABLE_NAME, null, values);
        db.close();

    }

    public String[]
SelectAllDataNamaIngredients(){
        SQLiteDatabase db;
        db = this.getReadableDatabase();

        String strSQL = "SELECT nama FROM
tabel_ingredients group by id_ingredients";
        Cursor cursor = db.rawQuery(strSQL,
null);

        String arrData[] = new
String[cursor.getCount()];
        if (cursor != null) {
            if (cursor.moveToFirst()) {
                for (int cc = 0; cc <
cursor.getCount(); cc++) {
                    arrData[cc] =
cursor.getString(cursor.getColumnIndexOrThrow(Da
tabaseHelper.KEY_NAMA));
                    cursor.moveToNext();
                }
            }
        }
        cursor.close();
    }

```

```

        return arrData;
    }

    public void importData() {
        try {
            for (int k = 1; k <= 7; k++) {

                Document doc =
Jsoup.connect("http://www.muslimconsumergroup.co
m/e-numbers_list.html?panna=" + k).get();
                dbHelper = new
DatabaseHelper(mContext);

                Elements tableElements =
doc.select("table[id=list_table]");
                Elements tableRowElements =
tableElements.select(":not(thead) tr");
                // String s[] = new
String[rows.size()];

                for (int i = 1; i <
tableRowElements.size(); i++) {
                    Element row =
tableRowElements.get(i);
                    Elements rowItems =
row.select("td");
                    Elements image =
row.select("img");
                    String url =
image.attr("title");

                    String id_ingredients =
row.child(1).text();
                    String nama =
row.child(2).text();
                    String keterangan =
row.child(3).text();
                    String kategori = url;
                    list.add(url);

                    System.out.println(id_ingredients);
                    System.out.println(nama);

                    System.out.println(kategori);

```

```

System.out.println(keterangan);

        addData(id_ingredients,
nama, kategori, keterangan);

        for (int j = 1; j <
rowItems.size(); j++) {

list.add(rowItems.get(j).text());

        }

    }

    } catch (IOException e) {
        e.printStackTrace();
    }

}

}

```

ObjectModel.java

```

public class ObjectModel {
    String word, meaning;

    public ObjectModel(String word, String
meaning) {

        this.word=word;
        this.meaning = meaning;

    }

    public String getWord()
    {
        return word;
    }

    public String getMeaning()
    {

```

```

        return meaning;
    }

```

RecyclerTouchListener.java

```

public class RecyclerTouchListener implements
RecyclerView.OnItemTouchListener {

    private GestureDetector gestureDetector;
    private ClickListener clickListener;

    public RecyclerTouchListener(Context
context, final RecyclerView recyclerView, final
ClickListener clickListener){
        this.clickListener = clickListener;
        gestureDetector = new
GestureDetector(context, new
GestureDetector.SimpleOnGestureListener(){
            @Override
            public boolean
onSingleTapUp(MotionEvent e){
                return true;
            }
            @Override
            public void onLongPress(MotionEvent
e){
                View child =
recyclerView.findChildViewUnder(e.getX(),
e.getY());
                if (child != null &&
clickListener !=null){
                    clickListener.onLongClick(child,
recyclerView.getChildPosition(child));
                }
            }
        });
    }

    @Override
    public boolean
onInterceptTouchEvent(RecyclerView rv,
MotionEvent e) {

```



```

        View child =
rv.findChildViewUnder(e.getX(), e.getY());
        if (child != null && clickListener !=
null && gestureDetector.onTouchEvent(e)) {
            clickListener.onClick(child,
rv.getChildPosition(child));
        }
        return false;
    }

    @Override
    public void onTouchEvent(RecyclerView rv,
MotionEvent e) {

    }

    @Override
    public void
onRequestDisallowInterceptTouchEvent(boolean
disallowIntercept) {

    }

    public interface ClickListener {
        void onClick(View view, int position);

        void onLongClick(View view, int
position);
    }
}

```

CustomAdapter.java

```

public class CustomAdapter extends
RecyclerView.Adapter<CustomAdapter.MyViewHolder>
{

    private ArrayList<ObjectModel> dataSet;
    Boolean check=false;
    public static class MyViewHolder extends
RecyclerView.ViewHolder {

```

```

        TextView word,meaning;

        RelativeLayout expandable;

        public MyViewHolder(View itemView) {
            super(itemView);
            this.expandable=
(RelativeLayout)itemView.findViewById(R.id.expan
dableLayout);
            this.word=
(TextView)itemView.findViewById(R.id.wordtext);
            this.meaning = (TextView)
itemView.findViewById(R.id.meaningtext);

        }
    }

    public CustomAdapter(ArrayList<ObjectModel>
data) {
        this.dataSet = data;
    }

    @Override
    public MyViewHolder
onCreateViewHolder(ViewGroup parent, int
viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate
(R.layout.card_view_row, parent, false);

        final MyViewHolder myViewHolder = new
MyViewHolder(view);
        view.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(!check)
                {

myViewHolder.expandable.animate()
                    .alpha(0.0f)
                    .setDuration(1000);

```

```

myViewHolder.expandable.setVisibility(View.GONE)
;

                check=true;
                //
myViewHolder.schedule.setVisibility(View.VISIBLE
);

                }
                else {

myViewHolder.expandable.setVisibility(View.VISIB
LE);

myViewHolder.expandable.animate()
                .alpha(1.0f)
                .setDuration(1000);

                check=false;

                }
        }
    });

    return myViewHolder;
}

@Override
public void onBindViewHolder(final
MyViewHolder holder, final int listPosition) {

    TextView word1= holder.word;
    TextView meaning1 = holder.meaning;

word1.setText(dataSet.get(listPosition).getWord(
));

meaning1.setText(dataSet.get(listPosition).getMe
aning());

}

@Override

```

```

        public int getItemCount() {
            return dataSet.size();
        }
    }
}

```

ListIngredients.java

```

public class ListIngredients extends
AppCompatActivity {

    private static RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager
layoutManager;
    private static RecyclerView recyclerView;
    public static ArrayList<ObjectModel> data;
    DatabaseHelper db ;
    ArrayList<String> wordcombimelist;
    ArrayList<String> meancombimelist;
    LinkedHashMap<String,String> namelist;
    SearchView searchView;
    protected Cursor cursor;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.list_ingredients);

        recyclerView = (RecyclerView)
findViewById(R.id.my_recycler_view);
        recyclerView.setHasFixedSize(true);
        db= new DatabaseHelper(this);
        db.importData();
        searchView = (SearchView)
findViewById(R.id.searchView);
        searchView.setQueryHint("Search Here");

        searchView.setQueryRefinementEnabled(true);
        layoutManager = new
LinearLayoutManager(this);
    }
}

```

```

recyclerView.setLayoutManager(layoutManager);
recyclerView.setItemAnimator(new
DefaultItemAnimator());
data = new ArrayList<ObjectModel>();
fetchData();

searchView.setOnQueryTextListener(new
SearchView.OnQueryTextListener() {
    @Override
    public boolean
onQueryTextSubmit(String query) {return false;
}

    @Override
    public boolean
onQueryTextChange(String newText) {

        newText = newText.toLowerCase();

        final ArrayList<ObjectModel>
filteredList = new ArrayList<ObjectModel>();

        for (int i = 0; i <
wordcombimelist.size(); i++) {

            final String text =
wordcombimelist.get(i).toLowerCase();
            if (text.contains(newText))
{

                filteredList.add(new
ObjectModel(wordcombimelist.get(i),meancombimeli
st.get(i)));
            }
        }
        adapter = new
CustomAdapter(filteredList);
recyclerView.setAdapter(adapter);

        return true;
}

```

```

        });
    }
    public void fetchData() {
        db =new DatabaseHelper(this);
        namelist=new LinkedHashMap<>();
        int ii;
        SQLiteDatabase sd =
db.getReadableDatabase();
        Cursor cursor =
sd.query("tabel_ingredients" ,null, null, null,
null, null, null);
        ii=cursor.getColumnIndex("nama");
        wordcombimelist=new ArrayList<String>();
        meancombimelist= new
ArrayList<String>();
        while (cursor.moveToNext()){
            namelist.put(cursor.getString(ii),
cursor.getString(cursor.getColumnIndex("keterangan"
an")));
        }
        Iterator entries =
namelist.entrySet().iterator();
        while (entries.hasNext()) {
            Map.Entry thisEntry = (Map.Entry)
entries.next();

            wordcombimelist.add(String.valueOf(thisEntry.get
Key()));
            meancombimelist.add("-
"+String.valueOf(thisEntry.getValue()));
        }

        for (int i = 0; i <
wordcombimelist.size(); i++) {
            data.add(new
ObjectModel(wordcombimelist.get(i),
meancombimelist.get(i)));
        }
        adapter = new CustomAdapter(data);
        recyclerView.setAdapter(adapter);

        sd.close();
    }
}

```

AboutUs.java

```

public class AboutUs extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.add_list_ingredients);

        String text = "HalalFinder by
Ingredients adalah aplikasi untuk memeriksa
kehalalan produk makanan dan minuman. " +
            "Aplikasi ini memeriksa produk
makanan dan minuman menggunakan ingredients yang
tertera pada produk yang ingin diperiksa. " +
            "Pengguna hanya memasukkan
ingredients yang tertera pada produk dan akan
memunculkan hasil berupa halal atau haram.";

        WebView webView =
(WebView) findViewById(R.id.webview);
        webView.loadData("<p style=\"text-align:
justify\">" + text + "</p>", "text/html", "UTF-8");

        webView.setBackgroundColor(Color.TRANSPARENT);

        String text2 = "Aplikasi ini menggunakan
database yang didapat dari European Economic
Community, " +
            "dan sudah diadopsi oleh
produsen makanan dan minuman di dunia. Jadi,
aplikasi ini dapat memberikan hasil " +
            "yang lebih akurat.";

        WebView webView2 =
(WebView) findViewById(R.id.webview2);
        webView2.loadData("<p style=\"text-
align: justify\">" + text2
+ "</p>", "text/html", "UTF-8");

        webView2.setBackgroundColor(Color.TRANSPARENT);
    }
}

```

LihatHasil.java

```

public class LihatHasil extends
AppCompatActivity {

    String daftar[];
    ListView listView;
    Menu menu;
    TextView textView;
    protected Cursor cursor;
    DatabaseHelper dbCenter;
    public static LihatHasil lihatHasil;
    private Context context;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_hasil);

        lihatHasil = this;
        dbCenter = new DatabaseHelper(this);
        RefreshList();
    }

    public void RefreshList() {
        SQLiteDatabase db =
dbCenter.getReadableDatabase();
        cursor = db.rawQuery("SELECT DISTINCT
nama_produk FROM tabel_hasil", null);
        daftar = new String[cursor.getCount()];
        cursor.moveToFirst();
        for (int cc=0; cc < cursor.getCount();
cc++){
            cursor.moveToPosition(cc);
            daftar[cc] =
cursor.getString(0).toString();
        }

        listView =
(ListView) findViewById(R.id.listview_hasil);
        textView =
(TextView) findViewById(R.id.emptytv);
    }
}

```



```

        listView.setEmptyView(textView);
        listView.setAdapter(new
ArrayAdapter(this,
android.R.layout.simple_list_item_1, daftar));
        listView.setSelected(true);
        listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void
onItemClick(AdapterView<?> parent, View view,
int position, long id) {
                final String selection =
daftar[position];
                final CharSequence[] dialogitem
= {"Lihat Hasil", "Hapus"};
                AlertDialog.Builder builder =
new AlertDialog.Builder(LihatHasil.this);
                builder.setTitle("Pilihan");
                builder.setItems(dialogitem, new
DialogInterface.OnClickListener() {
                    @Override
                    public void
onClick(DialogInterface dialog, int item) {
                        switch (item){
                            case 0:
                                Intent i = new
Intent(getApplicationContext(), LihatHasilCek.class);
                                i.putExtra("nama_produk", selection);
                                startActivity(i);

                                break;
                            case 1:
                                SQLiteDatabase
db = dbCenter.getWritableDatabase();

                                db.execSQL("delete from tabel_hasil where
nama_produk = '"+selection+"'");
                                RefreshList();
                                break;
                        }
                    }
                });
            }
        });

```

```

        builder.create().show();
    }
});

((ArrayAdapter) listView.getAdapter()).notifyData
SetInvalidated();
    db.close();
}
}

```

LihatHasilCek.java

```

public class LihatHasilCek extends
AppCompatActivity {
    protected Cursor cursor;
    DatabaseHelper dbHelper;
    String daftar[], daftar2[], daftar3[];

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.layout_hasil_ingredients
);

        dbHelper = new DatabaseHelper(this);

        LinearLayout myLayout =
        (LinearLayout) findViewById(R.id.linear_layout);
        LinearLayout.LayoutParams lp = new
        LinearLayout.LayoutParams(ActionBar.LayoutParams
.WRAP_CONTENT,
        ViewGroup.LayoutParams.WRAP_CONTENT);
        SQLiteDatabase db =
        dbHelper.getReadableDatabase();

        TextView txtNama =
        (TextView) findViewById(R.id.setNama);
        Button btnBack =
        (Button) findViewById(R.id.btnHasilBack);
        btnBack.setOnClickListener(new

```

```

View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});

String sql = "SELECT * FROM tabel_hasil
WHERE nama_produk = '"+
getIntent().getStringExtra("nama_produk") + "'
group by nama_ingredients";
cursor = db.rawQuery(sql,null);
daftar = new String[cursor.getCount()];
daftar2= new String[cursor.getCount()];
daftar3= new String[cursor.getCount()];
cursor.moveToFirst();
Log.i("sql = ",sql);

if (cursor.getCount()>0) {

    TextView[] textViews = new
TextView[cursor.getCount()];

    System.out.println(cursor.getCount());

    txtNama.setText(getIntent().getStringExtra("nama
_produk"));

    for (int cc = 0; cc <
cursor.getCount(); cc++) {
        cursor.moveToPosition(cc);
        daftar[cc] =
cursor.getString(cursor.getColumnIndex("nama_ing
redients"));
        daftar2[cc] =
cursor.getString(cursor.getColumnIndex("kode"));
        daftar3[cc] =
cursor.getString(cursor.getColumnIndex("keterangan"));

        textViews[cc] = new
TextView(this);
        textViews[cc].setTextSize(15);
    }
}

```

```

textViews[cc].setLayoutParams(lp);
        textViews[cc].setId(cc);

textViews[cc].setText(daftar[cc]+" , Kode E-
Number: "+daftar2[cc]+", "+daftar3[cc]);

textViews[cc].setPadding(0,10,0,10);
        myLayout.addView(textViews[cc]);
    }
}
db.close();
}
}

```

SearchUtama.java

```

public class SearchUtama extends
AppCompatActivity {

    private MultiAutoCompleteTextView mtv;
    private EditText namaProd;
    private DatabaseHelper dbHelper;
    private Context context;
    public SQLiteDatabase sqLiteDatabase;
    boolean kategori;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.search_utama);

        dbHelper = new DatabaseHelper(this);
        dbHelper.importData();
        Button btnSubmit =
(Button) findViewById(R.id.btnSubmit);
        btnSubmit.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                processData();
            }
        }
    }
}

```

```

    });

    Button btnLihatHasil =
    (Button) findViewById(R.id.btnLihatHasil);
    btnLihatHasil.setOnClickListener(new
    View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new
            Intent(getApplicationContext(), LihatHasil.class)
            ;

            startActivity(i);
        }
    });

    namaProd =
    (EditText) findViewById(R.id.tvNamaProduk);
    mtv =
    (MultiAutoCompleteTextView) findViewById(R.id.multiAuto);

    multiAutoComplete();
}
private void multiAutoComplete() {
    mtv =
    (MultiAutoCompleteTextView) findViewById(R.id.multiAuto);
    mtv.setTokenizer(new
    MultiAutoCompleteTextView.CommaTokenizer());

    final String[] namaIngredients =
    dbHelper.selectAllDataNamaIngredients();
    ArrayAdapter adapter = new
    ArrayAdapter(this,
    R.layout.support_simple_spinner_dropdown_item,
    namaIngredients);
    mtv.setAdapter(adapter);
    mtv.setThreshold(1);
    mtv.dismissDropDown();
}
public void processData() {
    String saveIngredients =
    mtv.getText().toString();
    String saveNamaProd =

```

```

namaProd.getText().toString();

        dbHelper = new DatabaseHelper(this);
        sqLiteDatabase =
dbHelper.getWritableDatabase();

        if (saveIngredients.isEmpty() ||
saveNamaProd.isEmpty()) {
            Toast.makeText(this, "Kolom isian
tidak boleh kosong !",
Toast.LENGTH_LONG).show();
        } else {

            if (saveIngredients != null &&
saveIngredients.length() > 0) {
                saveIngredients.substring(0,
saveIngredients.length() - 1);
                String ingredients[] =
saveIngredients.split(", ");

                int ya_tidak = 0;

                for (String s : ingredients) {
                    String sql = "SELECT
id_ingredients,nama,kategori,keterangan FROM
tabel_ingredients group by id_ingredients";
                    Cursor cursor =
sqLiteDatabase.rawQuery(sql, null);

                    while (cursor.moveToNext())
                    {
                        String str =
cursor.getString(cursor.getColumnIndexOrThrow("n
ama"));
                        String str1 =
cursor.getString(cursor.getColumnIndexOrThrow("k
ategori"));
                        String str2 =
cursor.getString(cursor.getColumnIndexOrThrow("i
d_ingredients"));
                        String str3 =
cursor.getString(cursor.getColumnIndexOrThrow("k
eterangan"));
                        String sqlInsert =

```

```

"INSERT INTO tabel_hasil
(nama_produk,nama_ingredients,kode,kategori,keterangan) VALUES ('" + saveNamaProd + "','" + s +
',' + str2 + "','" + str1 + "','" + str3 +
"');"

        if (s.equals(str)) {

System.out.println(str);

System.out.println(str1);

System.out.println(str2);

System.out.println(str3);

        try {

sqliteDatabase.execSQL(sqlInsert);
                                Log.i("sql = ",
sqlInsert);
        } catch (Exception
e) {

e.printStackTrace();

Toast.makeText(this, "Something wrong",
Toast.LENGTH_SHORT).show();
        }

        if

(str1.equals("haram") ||
str1.equals("unknown")){

                                ya_tidak = 1;
                                break;
        } else if

(!s.equals(str)){

                                ya_tidak = 2;
                                break;
        }
        }
    }

}

        sqliteDatabase.close();

```

```

        namaProd.setText("");
        mtv.setText("");

        if (ya_tidak == 1){
            //Toast.makeText(this,
            "Produk " + saveNamaProd + " keterangan Haram",
            Toast.LENGTH_SHORT).show();
            AlertDialog.Builder builder
            = new AlertDialog.Builder(this);

            builder.setTitle("Keterangan");
            builder.setMessage("Produk
            "+saveNamaProd+" Haram");

            builder.setPositiveButton("OK", new
            DialogInterface.OnClickListener() {
                @Override
                public void
            onClick(DialogInterface dialog, int which) {
                Log.e("info", "OK");
            }
            });
            builder.show();
        }
        else if (ya_tidak == 0){
            //Toast.makeText(this,
            "Produk " + saveNamaProd + " keterangan Halal",
            Toast.LENGTH_SHORT).show();
            AlertDialog.Builder builder
            = new AlertDialog.Builder(this);

            builder.setTitle("Keterangan");
            builder.setMessage("Produk
            "+saveNamaProd+" Halal");

            builder.setPositiveButton("OK", new
            DialogInterface.OnClickListener() {
                @Override
                public void
            onClick(DialogInterface dialog, int which) {
                Log.e("info", "OK");
            }
            });
            builder.show();
        }
    }
}

```



```

    }
    else if (ya_tidak == 2) {
        //Toast.makeText(this,
        "Produk " + saveNamaProd + " keterangan Halal",
        Toast.LENGTH_SHORT).show();
        AlertDialog.Builder builder
= new AlertDialog.Builder(this);

builder.setTitle("Keterangan");
        builder.setMessage("Terdapat
ingredients yang tak dikenali");

builder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void
onClick(DialogInterface dialog, int which) {
                Log.e("info", "OK");
            }
        });
        builder.show();
    }
}
}
}

```

“Halaman ini sengaja dikosongkan”

BIODATA PENULIS



Penulis bernama Haidar Ammarafif Ramzy, lahir di Suarabaya, 26 Januari 1995. Penulis merupakan anak pertama dari pasangan Bapak Afif Patriagama dan Ibu Fitri Ningsih. Sejak usia 4 tahun, penulis telah menempuh Pendidikan formal dimulai dari TK Suhardita Pisangan (1999-2001), SD Negeri Pisangan 1 Tangerang Selatan (2001-2007), MTs Negeri 3 Jakarta (2007-2010), dan SMA Negeri 108 Jakarta (2010-2013). Kemudian pada tahun 2013, penulis melanjutkan studi ke jenjang S1 di Departemen Matematika Institut Teknologi Sepuluh Nopember (ITS) Surabaya melalui jalur Mandiri dengan NRP 1213 100 117. Di Departemen Matematika, penulis mengambil Bidang Minat Ilmu Komputer. Selama menempuh Pendidikan di ITS, penulis aktif berorganisasi di Himpunan Mahasiswa Matematika ITS sebagai Head of Internal Bureau (2015-2016) dan Coordinator Conceptor of Student Resource (2015-2016). Penulis juga aktif diberbagai kepanitiaan seperti Olimpiade Matematika ITS (OMITS) 2015.

Adapun informasi lebih lanjut mengenai Tugas Akhir ini dapat ditujukan ke penulis melalui email haidarammarafif26@gmail.com